



Proyecto Golem-Universum: Adivina la carta

Reporte Técnico

Luis Pineda Cortés

Ivan Vladimir Meza Ruiz

Esther Venegas Briones

Lisset Salinas Pinacho

Elia Pérez Pavón

Hayde Castellanos Vargas

Alejandra Chavarría Amezcua

Héctor Avilés Arriaga

Caleb Rascón Estebane

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas

Universidad Nacional Autónoma de México

Contenido

1	Introducción.....	3
1.1	Antecedentes.....	3
1.2	Objetivos y metas.....	6
2	Marco teórico.....	7
2.1	Modelos de diálogo.....	7
2.2	Arquitectura cognitiva.....	9
3	El sistema Adivina la Carta.....	12
3.1	El escenario de Adivina la Carta.....	12
3.2	Modelos de diálogo.....	14
3.3	Módulos implementados.....	19
4	Recursos.....	21
4.1	Corpus de Mago de Oz.....	21
4.2	Corpus DIMEx100 niños.....	21
4.3	Corpus de interacciones.....	22
5	Evaluación.....	24
5.1	Primera evaluación.....	24
5.2	Segunda evaluación.....	25
5.3	Evaluación integral.....	27
6	Conclusiones y resultados.....	29
6.1	Productos de investigación.....	30
6.2	Difusión en medios.....	32
6.3	Agradecimientos.....	33
	Apéndice A: Participantes.....	35
	Apéndice B: Modelos de diálogos.....	36

1 Introducción

El proyecto Golem-Universum surge con el propósito de acercar al público general del museo de las ciencias Universum los avances más recientes del campo de la Inteligencia Artificial. Para cumplir con este propósito el grupo Golem del Departamento de Ciencias de la Computación (DCC) del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), en conjunto con el museo Universum, propusieron la construcción de un sistema conversacional que interactuara con niños que visitaran al museo para resolver una tarea. La tarea elegida fue la del juego “Adivina la Carta”. En este juego el niño debe interactuar con el sistema para adivinar la carta que éste eligió. La interacción del juego se definió como sigue: Golem escoge una carta entre 10 con motivos astronómicos, el niño tratará de adivinarla haciendo preguntas sobre las características del objeto representado en la carta. Golem responde cada una de las cuatro preguntas que el niño tiene oportunidad de formular. Después deberá decidirse por una carta y mostrársela al sistema que confirmará si efectivamente fue la que eligió.

El sistema desarrollado se denominó *Adivina la carta*. Constituye un sistema conversacional con entrada y salida multimodal. Durante el juego, el usuario interactúa con el sistema usando lenguaje hablado, pero además el sistema es capaz de interpretar imágenes a través de visión computacional, esto es para “ver” la carta que le muestra el niño. El sistema también se apoya del despliegue de imágenes y animaciones en la pantalla para reforzar la comunicación a través del lenguaje natural. Actualmente, el sistema funciona de manera permanente en una cabina construida específicamente con este fin en el museo Universum donde principalmente, niños entre 10 y 16 años de edad pueden jugar con Golem.

El presente reporte resume las características principales del sistema *Adivina la Carta* así como los resultados más importantes del proyecto. En esta sección se presentan los antecedentes, los objetivos, las metas y los elementos tecnológicos que conforman los sistemas conversacionales. En la sección 2 se presentan los modelos de diálogo y la arquitectura cognitiva, que constituyen el marco teórico al amparo del cual se desarrolló el sistema. En la sección 3 se presentan las características de la especificación y de la implementación del sistema. En la sección 4 se describen los recursos computacionales y lingüísticos recopilados, tanto para construir el sistema como para mejorarlo. La sección 5 se ocupa el tema de la evaluación del sistema; y finalmente, en la sección 6 se presentan las conclusiones y resultados del proyecto así como el trabajo a futuro.

A continuación, abordamos los antecedentes del proyecto y los objetivos y metas propuestos.

1.1 Antecedentes

Los sistemas conversacionales, como el sistema *Adivina la Carta*, son sistemas basados en el procesamiento del habla que se desarrollan en el campo de la Inteligencia Artificial, desarrollados con la finalidad de resolver una tarea específica. En contraste, hoy en día, para interactuar con los sistemas más comunes, el usuario tiene que aprender a utilizar una interfaz o un lenguaje de programación, y así resolver una tarea

Los sistemas conversacionales han evolucionado a través del tiempo; en los primeros sistemas la

interacción se basó en el lenguaje natural escrito a través del teclado [WEIZ66], [ALLEN06].

En la investigación y desarrollo de sistemas conversacionales predomina la que se realiza en idioma inglés; sin embargo, con el fin de estudiar y desarrollar sistemas conversacionales específicos para el español de México surgió en el Departamento de Ciencias de la Computación (DCC) del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS) el Proyecto DIME: Diálogos Inteligentes Multimodales en Español. Entre los resultados de este proyecto estuvieron:

- la recopilación y transcripción del Corpus DIMEx100 que es un corpus de grabaciones y análisis fonéticos del habla de la ciudad de México, cuyas propiedades permiten obtener modelos acústicos con un número relativamente pequeño de grabaciones y que se puede utilizar para el entrenamiento de modelos acústicos de un reconocedor de voz [PINEDA09].
- el desarrollo de una gramática computacional del español,
- el desarrollo del concepto “modelos de diálogo”,
- el desarrollo de un programa intérprete para la creación de sistemas conversacionales multimodales,
- la integración de una plataforma flexible para la creación de sistemas de reconocimiento de voz en español con español hablado y
- la investigación en la integración de eventos visuales en la interacción entre el sistema y el usuario [AGUILAR09]. En particular, el sistema Adivina la Carta usa esta modalidad adicional para “ver” la carta que el niño presenta como la correcta.

Para la creación del Proyecto Golem en 2002, se conjuntaron la experiencia y los resultados del proyecto DIME; se integró la tecnología de sistemas conversacionales y de reconocimiento de voz con distintas modalidades de entrada y de salida, así como las imágenes y el video, junto con la conducta motora del robot para la construcción de un agente móvil e inteligente que interactuara en el contexto de una conversación multimodal esquemática. En su versión inicial el robot Golem era capaz de sostener una conversación sencilla y de seguir comandos simples de movimiento.

Desde entonces, para el grupo Golem la integración y conformación de sistemas conversacionales es un tema de investigación que se ha venido desarrollando [AGUILAR09]. En junio de 2007 el robot Golem se presentó en Universum. Esta versión de Golem era capaz de dar una visita guiada al Departamento de Ciencias de la Computación del IIMAS mediante la explicación de una serie de pósters relativos a los proyectos del Departamento y de interactuar visualmente con los usuarios mediante el reconocimiento de gestos hechos con el brazo.

Con esta experiencia como antecedente se planteó el objetivo de tener un sistema permanente en el museo Universum con el cual los visitantes pudieran interactuar y acercarse a los avances de la Inteligencia Artificial. En 2009, con la incorporación de la capacidad de interpretación visual se presentó el módulo “Adivina la carta: Golem en Universum”. El sistema Adivina la Carta fue el primer sistema conversacional que desarrolló el Grupo y que abandonó el ambiente del laboratorio para ubicarse en un ambiente real. A partir de esta fecha, esta aplicación se exhibe de manera permanente en este museo, donde el público general, y especialmente los niños, pueden jugar a adivinar la carta que elige el sistema.

La tecnología involucrada en el desarrollo de sistemas conversacionales es diversa. En general, los elementos tecnológicos que conforman los sistemas conversacionales se representan en tres fases: el entendimiento del lenguaje, la administración del diálogo y la generación del lenguaje. La Figura 1 ilustra estas tres etapas.



Figura 1: Etapas de un sistema de diálogo

En la primera etapa, el reconocedor de voz tiene el objetivo de asignar una representación computacional a la señal acústica emitida por el usuario y transcribirla a una secuencia de palabras que represente lo dicho por el usuario de manera automática. Por ejemplo, si el usuario pregunta “¿la carta tiene un objeto de color rojo?” el sistema debe traducir esta emisión a un código computacional como “es(objeto,rojo)”. Posteriormente, la transcripción de lo dicho por el usuario es analizada por un parser semántico, cuyo objetivo es identificar una representación del significado de lo dicho. Un proceso similar ocurre en las interacciones que utilizan modalidades distintas a la del lenguaje. Por ejemplo, si un usuario hace un gesto para detener a un robot, éste debe traducirse en el código “parar”. El Grupo Golem cuenta con experiencia en la investigación y desarrollo de reconocedores automáticos de voz, en la sección 4 se presenta el corpus DIMEx100 niños recolectado en el contexto del proyecto Golem-Universum que replica al corpus DIMEx100 pero en este caso para voces infantiles que son los usuarios a quienes va dirigido principalmente el sistema Adivina la Carta.

En la etapa de administración del diálogo, el objetivo es llevar un control del estado de la conversación y decidir cuál es la mejor respuesta para el usuario. La intención es que la tarea del sistema se modele como un protocolo de interacción, en donde el sistema visite situaciones distintas para las cuales tiene ciertas expectativas que el usuario debe cumplir de manera particular. Una vez que lo dicho o hecho por el usuario coincide con las expectativas, se realiza la acción asociada a la expectativa, el sistema avanza para ubicarse en la siguiente situación. Con este marco se modeló la tarea para jugar a adivinar la carta. Para el ejemplo de la pregunta anterior, el sistema tiene que verificar si el objeto en la carta efectivamente es de color rojo, de serlo, debe contestar de forma afirmativa. Con respecto a esta etapa, el grupo Golem ha desarrollado un marco teórico [PINEDA08][PINEDA10], cuya propuesta se presenta a detalle en la sección dos de este documento.

Finalmente, en la etapa de generación del lenguaje, se realiza y produce un mensaje congruente

con la decisión tomada en la etapa anterior. El Grupo Golem utiliza tecnología estándar con dos componentes principales: un sintetizador de voz y la pantalla de despliegue. El sintetizador de voz ejecuta la tarea inversa del reconocedor de voz, transforma una transcripción de palabras en una señal acústica que corresponda a lo escrito. La pantalla se encarga de desplegar imágenes y animaciones en la pantalla que refuerzan de manera gráfica lo dicho por el sistema o que representan el estado en que se encuentra el sistema. Por ejemplo, si el sistema tiene una duda, puede desplegar unos signos de interrogación para acompañar una frase que indique que tiene una duda. Para nuestro ejemplo, una realización es que el sistema responda “*efectivamente, el objeto en la carta es de color rojo*”.

Golem ha causado un interés y una cobertura considerable por parte de los medios mexicanos; la televisión, el radio, la prensa escrita y el Internet y ha sido presentado en diversos eventos académicos. Algunos ejemplos de esta cobertura se presentan en el capítulo 6.2.

1.2 Objetivos y metas

El objetivo principal del proyecto Golem-Universum fue acercar a los visitantes del museo los avances recientes en el campo de Inteligencia Artificial. Con este objetivo en mente, se definieron los objetivos siguientes:

1. Implementar un sistema conversacional con el cual los visitantes del museo, especialmente los niños, puedan interactuar y aproximarse, a través del juego, a la inteligencia artificial.
2. Incrementar la robustez de los sistemas utilizados por el Grupo Golem.
3. Generar de productos de investigación como la presentación de artículos en congresos internacionales.
4. Generar recursos humanos relacionados con el desarrollo de sistemas conversacionales

Las tareas para lograr estos objetivos se definieron las tareas siguientes:

1. Desarrollo de módulos para integrar el sistema Adivina la carta.
2. Modelación de la tarea.
3. Integración del corpus *DIMEx100 niños* y creación de un reconocedor para la voz infantil en español de México.
4. Creación de un registro de interacciones.
5. Evaluación de la percepción de los usuarios del sistema Adivina la Carta.

2 Marco teórico

En esta sección se presenta el estado de la metodología para la especificación de sistemas inteligentes multimodales. Esta metodología permite el desarrollo de aplicaciones fijas y móviles, que requieran interactuar a través del lenguaje hablado, la visión y demás modalidades de entrada y salida. Es bajo esta metodología que se desarrolló el sistema Adivina la Carta.

Las subsecciones están organizadas de la siguiente manera. La subsección 2.1 presenta a los modelos de diálogo que es el mecanismo principal para la modelación de los protocolos de interacción entre usuario y máquina. En la subsección 2.2 se presenta la arquitectura cognitiva que establece la forma en que los modelos de diálogo se interrelacionan con los procesos de interpretación y generación para producir un ciclo de interacción entre usuario y sistema.

2.1 Modelos de diálogo

Los protocolos de interacción entre usuario y sistema se denominan modelos de diálogo y son centrales en la metodología de creación de los sistemas conversacionales. Un modelo de diálogo se define en términos de situaciones conversacionales. Una situación es una abstracción del estado del sistema que está formada por una situación inicial, una expectativa, las acciones a realizar cuando la expectativa se cumple y de la o de las situaciones a la que se arriba una vez realizada la acción.

Los modelos de diálogo tienen una representación gráfica donde las situaciones son representadas con nodos y las relaciones entre las situaciones con arcos. Cada arco tiene una etiqueta con la forma $\alpha:\beta$ donde α representa la expectativa y β la acción que el sistema ejecuta cuando la expectativa α es satisfecha para la situación actual s_i . Como resultado de ejecutar dicha acción el sistema arriba a la situación s_j . Esta situación se ilustra en la Figura 2.

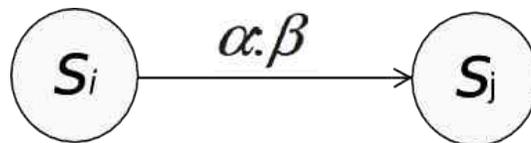


Figura 2: Representación gráfica de modelo de diálogo

Las expectativas pueden ser intencionales, es decir expresadas por el interlocutor, o naturales, es decir que representan eventos que pasan en el mundo. Por ejemplo, en el juego Adivina la Carta el sistema pregunta el nombre al usuario y entonces arriba a la situación donde espera escuchar un nombre; también puede estar en la situación en la que acaba de escoger la carta a adivinar, entonces esperará escuchar una pregunta sobre alguna características de la misma.

Cuando la expectativa cuenta sólo con un arco de salida de la situación s_i y la acción β se ejecuta de manera determinística; pero la expectativas α y las acciones β también pueden estar vacías (i.e., ϵ). En el caso de una acción vacía, el sistema se mueve a la situación s_j sin ejecutar ningún acto. Las expectativas también pueden ser concretas; por ejemplo, si el sistema se encuentra en una situación de hacer la oferta *¿quieres que te explique las instrucciones?* cuyas expectativas

son aceptar o rechazar, éstas se pueden representar por dos constantes (e.g., *si* o *no*) o dos predicados aterrizados (e.g., *answer(si)* o *awnser(no)*).

De hecho, la β corresponde a una estructura retórica que está compuesta por un número de acciones básicas. Esta definición de las acciones se basa en la Teoría de Estructuras Retóricas (RST, por sus siglas en inglés)[MANN88].

Cuando el sistema no puede interpretar la interacción del usuario, inicia un diálogo de recuperación con el fin de situar nuevamente al agente en la conversación. Las estrategias de recuperación son esenciales en situaciones donde la interpretación no es suficientemente robusta, por ejemplo en el caso del reconocimiento de voz.

La especificación de las expectativas y acciones es abstracta y no tiene relación con la modalidad en que la intención es expresada ni con la expresión concreta utilizada para su realización. Por ejemplo, en el caso del juego Adivina la Carta existe la posibilidad de que el usuario adivine la carta antes de hacer las cuatro preguntas permitidas de manera verbal, o puede realizar todas las preguntas y, al final, mostrar la carta a la cámara del sistema, es decir la expectativa y la realización pueden ser verbales o visuales.

La representación de las expectativas y las acciones en términos de actos del habla, con sus parámetros correspondientes permite recuperar la historia de la interacción a nivel intencional. Una vez que los arcos de una gráfica han sido atravesados, todas las expresiones representado las etiquetas de los arcos atravesados están aterrizados. Por lo tanto, la secuencia de situaciones y etiquetas para una iteración en particular constituye el discurso o el contexto anafórico, donde los predicados son intencionales (e.g., ofertar, ordenar, declarar,...) y los argumentos son las proposicionales expresadas y entendidas por el sistema en el dominio de la tarea.

En la versión del presente del formalismo, expectativas y acciones pueden ser especificadas con predicados, incluyendo variables libres. Por ejemplo, en la situación en la cual el sistema ha preguntado el nombre al usuario, se espera que el usuario satisfaga dicha expectativa proporcionando su nombre. En este caso la expectativa tiene la forma $\alpha = \text{nombre}(x)$ por lo que las variables libres de la expectativa se les tiene que asignar un valor que proviene de la percepción, esto provee al sistema con un componente de interpretación indexical. El usuario podría expresar su nombre de diferentes formas "*me llamo Juan*" o "*mi nombre es Juan*". Sin embargo, en todos estos escenarios el propósito es el asignar el nombre *Juan* como valor a la variable x . En este ejemplo el término indexical se usa en el sentido que las expresiones tienen valores en determinado tiempo, espacio, y en relación con el agente, es decir, en relación a los índices de la situación. Acciones también pueden ser especificadas como predicados con variables libres, por ejemplo la acción para la expectativa $\text{nombre}(x)$ en el contexto de una presentación puede ser $\beta = \text{saludo}(x)$, de tal forma que el arco: $\text{nombre}(x):\text{saludo}(x)$ para la realización de x como *Juan* el sistema podría producir un saludo como "*Hola Juan, mucho gusto en conocerte*". En este caso la variable es la misma, y una vez que es asignada a través de la percepción, su valor es ligado a la descripción de la acción. Adicionalmente, las situaciones también pueden ser expresadas como predicados con variable libres, de tal forma que los valores de una variable de una situación pueda estar ligada con un valor de la expectativa o de la acción que se atraviesa para llegar a ella.

Expectativas, acciones y las relaciones entre situación actual y siguiente pueden cambiar conforme la interacción con el usuario ocurre. Para modelar este fenómeno el formalismo permite especificar estos elementos como funciones. Estas funciones tienen como su principal argumento la historia de la interacción. Cada vez que una interacción llega a una situación, expectativa o acción especificada de manera funcional, las funciones son evaluadas primero y el resultado de esta evaluación sustituye al elemento correspondiente. La Figura 3 muestra la representación gráfica de las expectativas, acciones y siguientes situaciones funcionales. Para el caso de f la expectativa no está definida hasta llegar a la situación S_i en donde la función se evalúa. Esto es útil por ejemplo cuando se dan opciones al usuario, y la siguiente vez que se llega a la misma situación se deben omitir las opciones antes escogidas por el usuario, por lo tanto el conjunto de opciones esperadas es dinámico conforme avanza la conversación. La función g y h son evaluadas hasta el momento que se decide atravesar el arco en el que se encuentran. La función g se evaluará en una acción adecuada, por ejemplo si el arco corresponde a decir el mensaje de cuantas oportunidades le quedan al usuario, el mensaje cambiará con respecto al número de preguntas que haya hecho. De manera similar, la función h evaluará una situación. Por ejemplo, si al usuario le quedan oportunidades, el sistema debe ir a la situación de continuar con las preguntas, pero si no es el caso debe ir a la situación donde el usuario le muestra la carta adivinada.

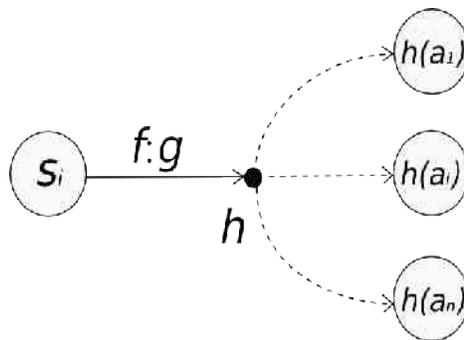


Figura 3: Expectativa, acción y situación siguiente funcionales

Las funciones pueden tener acceso al conocimiento del dominio específico o general. El conocimiento en este nivel es representado de manera independiente de la modalidad, es proposicional y es altamente independiente de la percepción. En analogía con la noción en psicología, esta memoria está almacenada en la *memoria semántica* del sistema.

Aun cuando las expectativas y las acciones son especificadas en una modalidad independiente, las situaciones tienen *tipos* de acuerdo con la modalidad perceptual. Las dos modalidades básicas son para escuchar (*listening*) y ver (*seeing*). El sistema puede ser fácilmente extendido con tipos de otras modalidades asociadas a otros dispositivos (e.g., botones, lasers, sonares, etc.) o con una combinación de modalidades.

Adicionalmente a las modalidades orientadas a coleccionar y interpretar la información de la entrada, la especificación de los modelos incluye una situación recursiva (*recursive*), que representa a todo un modelo de diálogo. Cuando una situación de este tipo es alcanzada, el diálogo actual es puesto en una pila (*push*), y el diálogo embebido se prepara para iniciar su interpretación en la situación inicial. Esta abstracción permite la construcción de grandes aplicaciones de forma modular y simple. Cuando un modelo alcanza una situación final el

modelo más saliente, si existe uno, es sacado (*pop*) y su interpretación continua para la situación recursiva, en donde las expectativas corresponden al mensaje de salida de modelo recientemente terminado. Este mensaje de salida es codificado en la situación final del modelo de diálogo.

2.2 Arquitectura cognitiva

La arquitectura cognitiva es el mecanismo que define como relacionar las interacciones del usuario con las expectativas del sistema dada una situación. Además define como producir y realizar las acciones producto de la expectativa asignada. La arquitectura tiene tres niveles conceptuales que se dominan: *representación/inferencia*, *interpretación/especificación* y *reconocimiento/generación*, además de la memoria semántica y la memoria perceptual, como se presenta en Figura 4. Esta arquitectura está orientada a la interacción. Existe un ciclo principal de interacción: reconocimiento → interpretación → representación → inferencia → especificación → generación. El nivel de *representación/inferencia* corresponde a los modelos de dialogo descritos en la sección anterior y especificados con el formalismo (F-RTN).

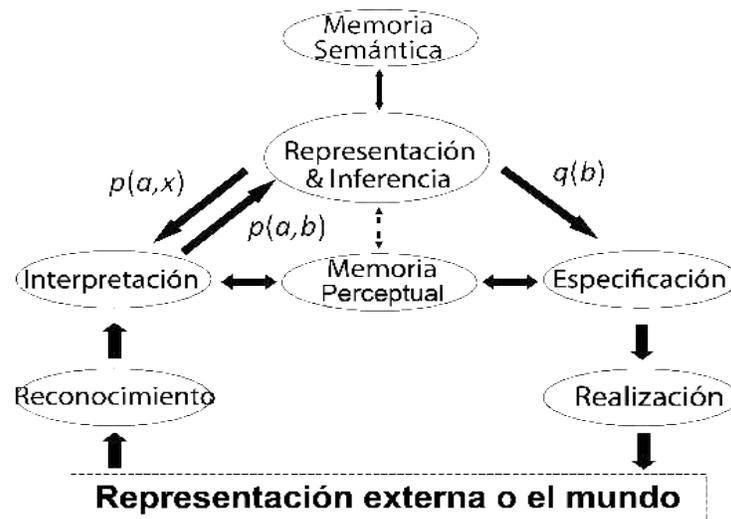


Figura 4: Arquitectura cognitiva

El reconocimiento y la interpretación corresponden en los niveles bajo y alto de la percepción. El reconocimiento es un proceso de abajo hacia arriba que traduce la información de entrada que corresponde a una interacción del usuario en una imagen o patrón en un formato codificado en relación a un algoritmo particular de reconocimiento. Sin embargo, esta imagen o patrón no tiene un significado o interpretación asignada. Estos patrones son específicos a la modalidad que corresponde a representaciones externas (interacciones) o al mundo mismo. Ejemplos, de los reconocedores son el reconocedor automático de voz que traduce la voz en una secuencia de palabras, sin interpretación. En el caso de visión, una imagen tomada por la cámara del sistema puede ser procesada con el algoritmo SIFT [LOWE04], el cual codifica la imagen en un conjunto de vectores de dimensionalidad superior, donde cada punto del vector representa un punto saliente que es invariante a la rotación y traslación.

Una implementación particular de esta arquitectura tiene un número finito de formatos de

codificación. Cada formato es asociado con un mecanismo de reconocimiento. En nuestra terminología, cada uno de estos formatos representa a una “*modalidad*”. En el caso de la generación, los mecanismos van a transformar las imágenes en representaciones externas. Ejemplos, de mecanismos de generación son el sintetizador de voz que traduce una oración escrita en mensaje de voz y el despliegue de dibujos en la pantalla.

En el nivel de interpretación de la arquitectura, corresponde a la percepción de alto nivel. Este nivel tiene el objetivo de asignar interpretaciones a los patrones sin interpretación, dada la situación del modelo específico siendo ejecutado. El componente central es el programa intérprete de modelos de diálogo. Las entradas para el interprete son las expectativas del agente dada una situación con un flujo de arriba hacia abajo en la arquitectura y las imágenes sin interpretación con un flujo de abajo hacia arriba. Las salidas del interprete son interpretaciones. Los eventos del mundo también pueden ser interpretados en la misma forma. La figura 4 ejemplifica este proceso de interpretación. Ahí $p(a,x)$ representa la expectativa del agente en cierta situación, mientras que el intérprete dada una imagen sin interpretación y la expectativa misma asigna la interpretación $p(a,b)$. En este mensaje, la variable x es pasada a través del modelo diálogo (como se explicó en la sección anterior) y la acción $q(b)$ es ejecutada en el canal de salida.

La memoria perceptual es un conjunto de asociaciones entre interpretaciones y imágenes sin interpretación. Las imágenes almacenadas aquí pueden ser objetos complejos estructurados, pero en el caso más sencillo las interpretaciones corresponden directamente a un patrón sin interpretación. Las asociaciones almacenadas en esta memoria pueden ser recolectadas de manera previa, y los formatos utilizados para las imágenes corresponden a las modalidades específicas del sistema. El interprete que hace uso de esas asociaciones puede usar métodos cualitativos o de aproximación para identificar la asociación correcta. Por ejemplo, el interprete de lenguaje natural puede usar un análisis sintáctico para identificar el significado de la oración reconocida por el ASR, o el intérprete visual puede calcular que vector SIFT de los almacenados en memoria está más cercano al vector SIFT que capta la cámara. Adicionalmente, las asociaciones almacenadas en la memoria perceptual son accesibles desde el manejador de diálogo a través de las funciones, para situaciones en las que se requiera planeación, razonamiento, etc.

Una discusión más extensa de la especificación de los modelos de diálogos, la arquitecta cognitiva y los alcances se puede consultar en [PINEDA10]. Cabe destacar que esta publicación es un productos de investigación posibles a través de este proyecto.

3 El sistema Adivina la Carta

En esta sección se presentan los aspectos de especificación e implementación del sistema conversacional Adivina la Carta que se encuentra en el museo de las ciencias Universum funcionando de manera permanente. El sistema es capaz de interactuar con visitantes del museo para jugar el juego de adivina la carta. En este juego un participante, el sistema, escoge una carta de un grupo de diez y un segundo participante, el usuario, trata de adivinar la carta a través de una serie de preguntas relacionadas con las propiedades del objeto representado en la carta.

El juego tiene el propósito de acercar a los visitantes con los avances del campo de Inteligencia Artificial en particular con los sistemas conversacionales. Para cumplir con lo anterior, el sistema puede interactuar con el usuario a través del lenguaje hablado, pero también acepta interacciones visuales, por ejemplo, cuando el niño muestra la carta que cree que el sistema eligió a la cámara del sistema.

Las siguientes subsecciones están organizadas de la siguiente manera. En la subsección 3.1 se presenta el escenario del juego, tanto como las reglas establecidas como el escenario físico. En la subsección 3.2 se especifican los modelos de diálogo utilizados para el juego de adivina la carta, así como un ejemplo de interacción usando dicho modelo. Finalmente, en la subsección 3.3 se describen los módulos implementados, es decir: los reconocedores, los interpretes, los generadores y los realizadores que forman parte de la arquitectura cognitiva presentada en la subsección 2.2.

3.1 El escenario de Adivina la Carta

En el juego de Adivina la Carta, el objetivo del usuario es obtener la mayor cantidad de información posible, que le permita deducir la carta elegida por el sistema de un conjunto de diez cartas posibles. Para esto, el usuario puede plantear hasta cuatro preguntas, en español hablado, sobre las propiedades del objeto representado en la carta, tales como: *¿es de color rojo? ¿es redondo? ¿es un planeta?*

Los usuarios son, preferentemente, niños entre los diez y dieciséis años de edad; sin embargo, en el museo a todas las personas les es permitido interactuar con el sistema, no así a los niños menores de 10 años, ya que para ellos la tarea es considerablemente compleja. Las cartas tienen motivos astronómicos, ya que 2009 fue el año internacional de la astronomía¹. Los objetos escogidos para las diez cartas fueron: el Sol, la Luna, la Tierra, Júpiter, Mercurio, un telescopio, un radio-telescopio, una galaxia elíptica, una galaxia espiral y una nebulosa (ver Figura 5).

¹<http://www.astronomia2009.org.mx/>



Figura 5: Cartas del juego Adivina la Carta

Una sesión típica de la interacción sigue la siguiente estructura:

1. El sistema saluda y se presenta con el usuario
2. El sistema pregunta al usuario su nombre y edad.
3. El sistema pregunta al usuario si requiere escuchar las instrucciones del juego, en caso afirmativo el sistema las explica.
4. El sistema elige la carta.
5. La fase de preguntas comienza, en ella el usuario puede hacer hasta cuatro. Después de cada pregunta el sistema contesta al usuario si efectivamente la característica es parte del objeto de la carta.
6. El sistema pide al usuario muestre la carta que cree que el sistema eligió y el usuario la muestra.
7. El sistema dice al usuario si adivinó o no.
8. El sistema agradece al usuario su participación y se despide.

Golem se encuentra en un módulo permanente, ubicado en el museo Universum. El módulo está formado por una cabina, diseñada a propósito de este proyecto por personal del museo. Dentro de la cabina está todo el equipo necesario para el juego: en una mesa se localizan las diez cartas, para que el usuario las manipule a su gusto, un monitor conectado a una computadora personal oculta que despliega imágenes y otro tipo de información que facilita la comunicación con el usuario; un micrófono que utiliza el usuario para interactuar con el sistema de manera hablada; una cámara a través de la cual el sistema “ve”, dos bocinas que reproducen la voz del sistema y dos sillas. La Figura 6 muestra esta distribución.

La cabina es un espacio cerrado que tiene como fin adicional el de aislar, en la medida de lo posible, el sistema de ruidos externos que son comunes en la actividad diaria del museo. Sin embargo, una de sus paredes es transparente y en su parte exterior hay un monitor y otro par de bocinas que reproducen lo que el usuario ve y lo que el usuario y el sistema dicen, lo que permite a los visitantes seguir desde el exterior la interacción entre el usuario y el sistema. Esta distribución se puede observar en la Figura 7.



Figura 6: Vista interior de la cabina del sistema Adivina la Carta



Figura 7: Vista exterior de la cabina del sistema Adivina la Carta

3.2 Modelos de diálogo

La estructura de la tarea del juego Adivina la Carta requirió de ocho modelos de diálogo:

- El principal (*M*) que agrupa a todos los demás y siete subdiálogos encargados de acciones específicas de la tarea cuya estructura es la siguiente:
- Saludo (*G*): agrupa los subdiálogos que involucran el inicio del juego antes de las preguntas del usuario. También, en este subdiálogo el sistema elige de manera aleatoria la

carta.

- *Reconocimiento de nombre (n)*: el sistema pregunta el nombre del usuario, y de reconocerlo, lo saluda por su nombre.
- *Verificación de la edad (a)*: se pregunta la edad y se verifica que el usuario pueda jugar.
- *Instrucciones (i)*: se pregunta al usuario si requiere escuchar las instrucciones del juego y de ser así se dan las instrucciones.
- *Preguntas (I)*: el sistema requiere al usuario las preguntas, las escucha y contesta con una respuesta adecuada, dependiendo de la carta elegida.
- *Visualización (V)*: el usuario muestra la carta a la cámara, el sistema la “ve” y confirma si es o no la carta escogida.
- *Despedida (F)*: el sistema agradece al usuario su participación y se despide de él por su nombre, de ser posible.

En la Tabla 1 se presenta una interacción entre el usuario y el sistema Adivina la Carta utilizando esta estructura de subdiálogos. En la columna MD vemos los identificadores de los modelos de diálogos a los que pertenecen los turnos. En la tabla también se presentan las acciones gráficas del sistema así como las interacciones visuales del usuario (ver turno 28). Además, se presenta la hipótesis del módulo del reconocedor de voz (después del símbolo /) en los turnos del usuario y su interpretación correspondiente.

Tur	H.	MD	Elocución	Inter.	Acción visual
1	SYS	M	Hola a todos me llamo Golem y éste es el juego de adivina la carta, tenemos jugador, ¿Cuál es tu nombre?		Imagen de bienvenida
2	USR	n	Octavio/uno	notU	
3	SYS	n	Estoy medio sordo, ¿Cuál es tu nombre?		
4	USR	n	Octavio/hector luz	name(hector)	
5	SYS	n	Hola Hector, me da gusto jugar contigo		
6	SYS	a	¿Me puedes decir tu edad?		
7	USR	a	Trece años/trece años	age(13)	
8	SYS	a	¿Seguro que tienen trece años?		
9	USR	a	Sí/si	ok	
10	SYS	a	Muy bien, entonces vamos a jugar		
11	SYS	i	¿Quieres que te explique las reglas del juego?		
12	USR	i	Sí/si	ok	
13	SYS	i	Presta atención... después de la ronda de preguntas compararemos mi carta y la que crees que elegí. Veremos si ganaste ¿Entendiste las instrucciones?		
14	USR	i	Sí/si	ok	
15	SYS	i	Okey, fijate muy bien en las cartas mientras escojo uno		
16	SYS	I	Empecemos el juego, por favor hazme tu primera pregunta		
17	USR	I	¿Es roja?/alido	notU	
18	SYS	I	¿Me puedes preguntar otra característica que veas? Por favor		
19	USR	I	¿Se parece a un plato?/parece un plato	feat(dish)	
20	SYS	I	Lo siento, eso no está en mi carta. Has la siguiente pregunta		Imagen con un plato.
21	USR	I	¿Es redonda?/onda	feat(wave)	
22	SYS	I	No hay nada como una onda. Hazme la siguiente pregunta.		Imagen con ondas de agua.
23	USR	I	¿Es azul?/luz	feat(light)	
24	SYS	I	No le atinaste, en mi carta nada tiene luz propia. Okay, ésta es la última pregunta que me puedes hacer, piénsala muy bien.		Imagen con de luz
25	USR	I	¿Es rojo?/es rojo	feat(red)	
26	SYS	I	Veo algo parecido pero no es rojo		Imagen de color rojo
27	USR	V	Por favor muéstrame la carta		
28	USR	V	El usuario muestra la carta el sol a la cámara	card(sun)	
29	SYS	V	No tengo muy buena vista. ¿Es ésta la carta que me mostraste?		
30	USR	V	Sí/si	ok	
31	SYS	V	Lástima, la carta que había escogido es la del planeta mercurio		La carta de mercurio
32	SYS	F	Oh! Pero antes de irte por favor mira a la cámara. Gracias por jugar conmigo Hector, regresa pronto.		Cámara toma foto y la despliega

Tabla 1: Ejemplo de interacción entre el sistema Adivina la Carta y usuario (Tur, turno; H, hablante; MD, modelo de diálogo)

Las Figuras 8 y 9 ilustran a los modelos de diálogos principal (M) y de preguntas (I). La especificación completa de los modelos de diálogo se pueden encontrar en el apéndice B de este documento.

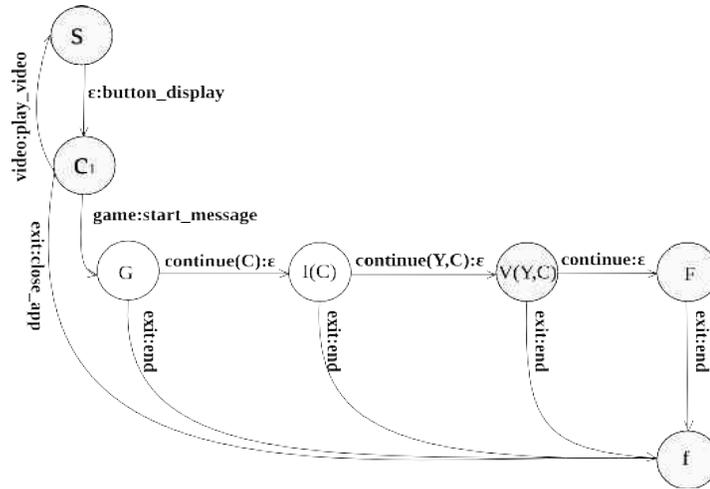


Figura 8: Representación gráfica del modelo principal (M) del sistema Adivina la Carta

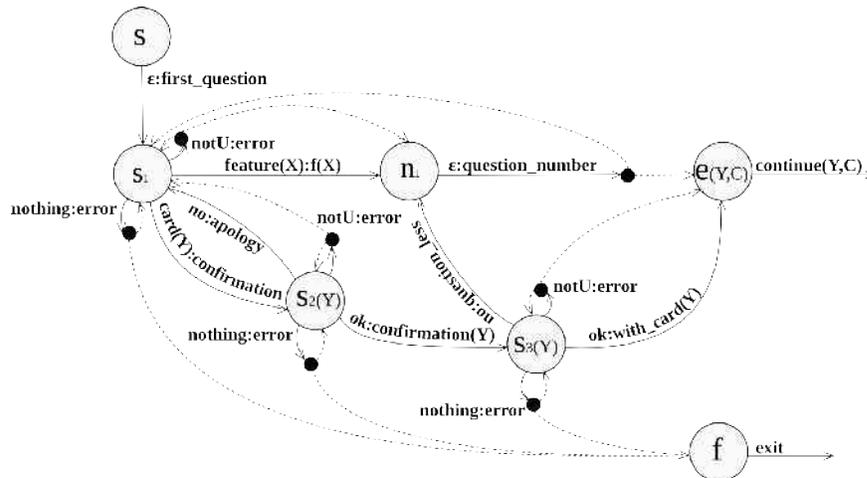


Figura 9: Representación gráfica del modelo de preguntas (I) del sistema Adivina la Carta

Una interacción típica comienza en la situación inicial S del diálogo principal (M , ver Figura 8). Como se explicó en la sección dos de este documento, los arcos indican las expectativas y las acciones, y además designan la siguiente situación. En la situación inicial, el único arco está etiquetado con el par $\epsilon:button_display$. Lo que significa que no hay una expectativa; la acción se ejecuta de manera determinística y se arriba a la situación C_1 . En este caso la acción consiste en desplegar una serie de botones en la pantalla que permiten al usuario escoger entre ver un video ilustrativo del sistema, terminar el juego o jugarlo. Estas opciones corresponden con las tres expectativas de la situación C_1 (i.e., *video*, *exit* y *game*). En el modelo principal también vemos cuatro situaciones recursivas que representan cuatro subdiálogos. La situación de saludo (G), de preguntas (I), de visualización (V) y la de despedida (F). Cuando el flujo de la interacción

alcanza estas situaciones el subdiálogo correspondiente se carga y ejecuta. Una vez que el subdiálogo termina de ejecutarse, al llegar a una situación final puede pasar al subdiálogo que lo ejecutó un mensaje que se convierte en la expectativa de la situación recursiva. En el modelo principal (M) las situaciones recursivas tienen dos expectativas: una es continuar con el juego (*continue*) o terminar (*exit*). El mensaje depende de lo que ocurre en el subdiálogo; si las interacciones del usuario son interpretadas correctamente el sistema continúa, sin embargo, si el usuario deja de interactuar con el sistema la interacción termina. Los mensajes de salidas pueden tener parámetros; para el caso del subdiálogo de saludo (G) la opción de continuar contiene el parámetro C que representa la carta escogida por el sistema, y que es pasado a la siguiente situación de preguntas (I) que se encarga de usar esta información para proporcionar la respuesta adecuada a las preguntas del usuario. El parámetro Y representa a la carta adivinada por el usuario antes de utilizar sus cuatro preguntas.

Una vez que se carga un subdiálogo a través de las situaciones recursivas, la interacción se sitúa en la situación inicial de dicho modelo de diálogo. Por ejemplo, en el caso del subdiálogo de preguntas (I) se sitúa en la situación S (ver Figura 9). En esta situación y de manera similar a lo que ocurre con la expectativa vacía en el diálogo principal, se atraviesa el único arco y se ejecuta la acción que consiste en hacer saber al usuario que puede hacer su primera pregunta y se llega a la situación S_1 . En esta situación se tienen dos expectativas, que el usuario haga una pregunta sobre alguna característica de los objetos en la carta (i.e., $feature(X)$) o que trate de adivinar la carta (i.e., $card(Y)$). En el primer caso el arco correspondiente es atravesado en donde la función $f(X)$ determina la acción a ejecutar; en este caso la acción depende de la propiedad por la que preguntó el usuario, representada por la variable X , por lo que no se puede especificar una acción de antemano. Sin embargo en el formalismo de F-RTN esto se puede capturar a través del uso de las funciones. Una situación similar se da para el arco $\epsilon:questions_number$ donde el sistema genera un mensaje adecuado con respecto al número de oportunidades que usuario tiene. Sin embargo, en este caso la siguiente situación también depende del número de oportunidades, si el usuario tiene suficientes oportunidades debe seguir escuchando preguntas, es decir regresar a la situación S_1 . Sin embargo, si ya no tiene más oportunidades el juego debe avanzar para que el usuario adivine la carta, es decir, ir a la situación final e . Como se observa en la figura, la función se representa por un punto y las situaciones a las que se puede ir por líneas punteadas (ver explicación asociada a figura 3).

Si el usuario decide adivinar la carta antes de consumir sus oportunidades el sistema atraviesa el arco $card(Y):confirmation$; para lidiar con los casos en que el sistema mal entiende una expresión del usuario, el sistema confirma que efectivamente el usuario está tratando de adivinar la pregunta (la situación S_2) y que la carta que escuchó el sistema es efectivamente la que trató de adivinar el usuario (la situación S_3); una vez que estas confirmaciones son positivas el modelo llega a la situación final e , que contiene los parámetros de la carta escogida (C) y la carta adivinada (Y), que serán utilizadas por el subdiálogo de visualización (V) del modelo de diálogo principal (ver Figura 8).

Adicionalmente los modelos de diálogos incluyen estrategias de recuperación para cuando ciertos errores ocurren. En particular, para el reconocimiento de voz se identificaron dos fuentes de error, cuando el reconocedor no escucha (representado por la expectativa *nothing*) y cuando no entiende lo dicho por el usuario (representado por la expectativa *notU*). En el modelo de las preguntas se puede apreciar estas situaciones (ver Figura 9), como se puede ver, las acciones y las situaciones siguientes de las expectativas dependen de la evaluación de una función. En este

caso, la función mide cuantas veces ha ocurrido este error para esta situación dada y en caso de ser demasiadas se decide terminar el diálogo o continuar.

Una discusión más extensa de la especificación de los modelos de diálogos y sus diferentes versiones puede encontrarse en [MEZA10a] y [MEZA10b]. Cabe destacar que ambas publicaciones son productos de investigación de este proyecto.

3.3 Módulos implementados

A continuación se describe cada uno de los módulos que integran al sistema Adivina la Carta.

Reconocedor de voz

El módulo del reconocedor de voz se encarga de transcribir la señal de voz del usuario en una transcripción ortográfica. Para la creación de un reconocedor de voz es necesario considerar tres partes del reconocedor:

- El modelo acústico: en esta parte se modela la relación entre la señal de audio correspondiente a la voz y las unidades fonéticas mínimas. Es decir, relaciona de señal de audio a fonema. Para crear este modelo, es necesario contar con un conjunto de ejemplos de grabaciones y sus transcripciones fonéticas. En particular, para capturar de manera amplia a los usuarios del sistema que son niños se recolectó en conjunto con el museo Universum el Corpus DIMEx100 niños (ver subsección 4.2).
- El diccionario: en esta parte se modela cómo las unidades fonéticas mínimas se conjuntan para formar palabras. El diccionario debe contener las palabras más comunes para la tarea. En particular, para el sistema Adivina la Carta, el modelo del lenguaje que se explica a continuación define el conjunto de palabras.
- El modelo de lenguaje: en esta parte se modela la relación entre las palabras para formar oraciones. En particular se modela que palabras siguen a que otras. Para crear este modelo es necesario crear ejemplos de oraciones/elocuciones posibles en la tarea. En particular, para el proyecto Golem-Universum existen varias versiones que se fueron perfeccionando conforme el proyecto avanzó (ver subsección 4.3).

Este módulo se basa en Sphinx [HUERTA99], en particular para su versión 3. El módulo se creó para voz espontánea del usuario, a diferencia de que el usuario leyera. Además, es para voz en vivo, a diferencia que lo dicho por el usuario fuera grabado, por lo que el sistema además debe ser capaz de identificar cuando el usuario está hablando.

Intérprete de lenguaje natural

El intérprete de lenguaje natural se encarga de identificar el significado de la interacción hablada del usuario, es decir su interpretación. Para esto toma la transcripción ortográfica que entrega el reconocedor de voz y la analiza. Como se explicó en la subsección 2.2 correspondiente a la arquitectura cognitiva, para realizar esta tarea el intérprete tiene también acceso a las expectativas dada una situación. La expectativa define la estrategia a utilizar para identificar el significado. Por ejemplo, si el sistema se encuentra en la situación de escuchar una pregunta del usuario, quiere decir que está a la expectativa de una pregunta sobre las características del objeto en la carta (i.e., *feature(X)*) o que el usuario intente adivinar la carta (i.e., *card(Y)*). En este caso, ambas expectativas definen a las estrategias para identificar la interpretación.

El intérprete se base en identificación de palabras claves. Sin embargo, en lugar de sólo buscar por palabras claves, el intérprete usa expresiones regulares lo que permite una mayor flexibilidad en los patrones lingüísticos a analizar. Cada expectativa tiene asociada un conjunto de expresiones regulares que le permiten identificar los argumentos de ésta. Por ejemplo, supongamos que en la situación del ejemplo anterior el usuario preguntó *¿es de color rojo?*. Dado que una de las expectativas es *feature(X)* utilizará las expresiones regulares asociadas a esta expectativa para identificar el valor de X. Dado que uno de los patrones incluye el patrón *color algo*, ese algo será identificado como el valor buscado. En este caso la asignación sería $X=red$ y la interpretación que regresa el interprete sería *feature(red)*. Esta situación se observa en el turno 25 de la Tabla 1.

Reconocedor visual

El módulo del reconocedor visual se encarga de tomar una foto a través de la cámara y obtener una representación de la imagen con base en puntos característicos. Es decir, codifica una imagen en términos de puntos que sean invariantes a la rotación y la escala. De esta forma, la imagen se reduce a sus puntos más importantes. El reconocedor actúa cuando el usuario muestra la carta a la cámara del sistema. Para generar dicha codificación usamos el algoritmo SIFT [LOWE04].

Intérprete visual

Este módulo encarga de tomar la codificación generada por el reconocedor visual y compararla con la codificación almacenadas en la memoria perceptual de las diez cartas posibles. La comparación hecha nos arroja qué carta se parece más a la de la imagen tomada por el reconocedor; de esta forma se toma la que más se parece a las “imágenes” guardadas. La codificación de la imagen en donde el usuario muestra la carta al sistema depende de la iluminación y los reflejos que esta produce, por lo que pueden haber errores. Para mitigar estos casos el modelo de diálogo confirma que efectivamente la carta identificada por el interprete visual coincida con la mostrada por el usuario.

Sintetizador de voz

El módulo de síntesis de voz se encarga de decir con voz hablada los mensajes que el sistema decida. Para los turnos marcados en Tabla 1 con el hablante como SYS las elocuciones corresponden a estos mensajes. Estos son creados dinámicamente. Para este módulo usamos el sistema Mbrola con una voz para español de Venezuela que se ha convertido en la voz del sistema Golem [DUTOIT96].

Despliegue de imágenes

El módulo se encarga de mostrar imágenes en la pantalla. Las imágenes tienen el propósito de complementar los mensajes sintetizados por el sistema. Las imágenes desplegadas fueron diseñadas por el personal del museo Universum y corresponden a la imagen gráfica diseñada para el sistema. Adicionalmente, este módulo también funciona como un intérprete de acciones gráficas (e.g, clicks).

Manejador de diálogo

El módulo del manejador de diálogo se encarga de ejecutar los modelos de diálogo para el sistema Adivina la Carta. El manejador corresponde al nivel de *representación/inferencia* de la arquitectura cognitiva presentada en la subsección 2.2. Este módulo además de ejecutar los modelos de diálogos construye la historia de la conversación y coordina los demás módulos. Para

el sistema Adivina la Carta se extendió el módulo para registrar las interacciones de tal forma que después se pudieran analizar estos registros y obtener información sobre el desempeño de todo el sistema.

Para la integración de los módulos se usa la arquitectura abierta de agentes (OAA, por sus siglas en inglés) [CHEYNER01]. Esta arquitectura la utilizamos como mecanismo de comunicación entre los módulos.

4 Recursos

Un proyecto como el juego Adivina la Carta requiere partir de evidencia empírica para modelar la tarea así como para su constante mejoramiento y evaluación. En este capítulo presentamos una serie de recursos que se crearon a partir del presente proyecto, útiles desde el diseño de la aplicación hasta su evaluación.

Inicialmente, para comenzar el análisis de la tarea, fue necesario conocer cómo los potenciales usuarios se comportarían y el tipo de lenguaje que utilizarían durante la interacción con el sistema. El escenario de Mago de Oz ayudó a cumplir estos objetivos. En nuestro caso, un miembro del equipo pretendió ser el sistema mientras interactuaba con potenciales usuarios. Con esta técnica se recopiló un corpus que fue con el que se realizó el primer análisis de la tarea. En la subsección 4.1 se presenta el corpus de Mago de Oz que se utilizó como la primera aproximación de la tarea.

Otro recurso importante es el reconocedor de voz, que en los sistemas conversacionales constituye muchas veces el cuello de botella para un óptimo desempeño, por lo que es necesario dedicar recursos explícitos para su adecuada creación. Para el sistema Adivina la Carta, dado que los usuarios eran niños, fue necesario recopilar un corpus de voces infantiles al cual denominamos *DIMEx100 niños*. Adicionalmente, para identificar el lenguaje que los niños usan al interactuar con el sistema, se grabaron y analizaron las interacciones de usuarios con el sistema. Este conjunto de registros Esta colección de grabaciones, conformó el corpus de interacciones, ha sido un recurso muy importante no sólo en el desarrollo del sistema, sino que tienen un gran potencial para futuras investigaciones. En la subsección 4.2 describiremos a detalle este recurso.

En la subsección 4.3 se presenta el corpus de interacciones que fue utilizado para analizar el lenguaje de los usuarios.

4.1 Corpus de Mago de Oz

Un ejercicio inicial en el desarrollo del proyecto Adivina la Carta fue analizar el comportamiento de los usuarios ante la tarea. Para lograrlo se optó por una variante relajada del escenario del Mago de Oz. En esta versión un miembro del equipo (el mago) actuó como el sistema e interactuó con los usuarios. En este ejercicio no se trató de engañar totalmente al usuario para hacerlo creer que interactuaba con un programa, sino que interactuara directamente con el mago. Las instrucciones del mago eran seguir un pequeño protocolo que reflejaba la tarea, tal y como una computadora lo hubiera hecho. Los objetivos de este ejercicio fueron:

- Crear un primer protocolo de la tarea, en particular, identificar el número de preguntas que necesitaba un niño para intentar adivinar la carta elegida por el sistema.
- Identificar el tipo de preguntas que los niños hacían.

Se realizaron veinte grabaciones de audio y video del mago jugando con los usuarios, las cuales se transcribieron y analizaron para determinar, entre otros temas, que el número apropiado de preguntas eran cuatro. Los resultados principales de este trabajo pueden consultarse en el reporte [HERNANDEZ09]; el cual se realizó en el contexto del proyecto y como parte del programa Jóvenes a la Investigación.

4.2 Corpus DIMEx100 niños

El desempeño del reconocedor de voz depende en gran medida de los modelos acústicos (ver subsección 3.3). Debido a que el sistema Adivina la Carta iba interactuar principalmente con niños se decidió recopilar un corpus de voces de niños. Dada la experiencia del grupo con el corpus DIMEx100 [PINEDA09] se decidió reproducir dicho corpus pero con voces infantiles con la ayuda del museo Universum. El corpus DIMEx100 consiste en grabaciones de 100 adultos de origen mexicano en donde cada sujeto lee 60 oraciones (50 únicas y 10 comunes a todos). La selección de las oraciones es tal que los fonemas están balanceados para el español, es decir tienen una distribución similar a la que se encuentra en el idioma español en particular para la variante hablada en México.

El corpus recolectado se le denomina DIMEx100 niños y consiste de 100 niños de origen mexicano, en su totalidad de la ciudad de México, en donde cada niño lee 50 oraciones únicas. Esto es una diferencia en comparación con el corpus original que además los usuarios leían 10 oraciones comunes. Dado a que en algunas ocasiones los niños tendían a cometer errores en la lectura o en la pronunciación de las palabras, las oraciones fueron verificadas para que reflejaran lo que los niños habían dicho. Las características generales de tanto los hablantes como las oraciones del corpus DIMEx100 niños se presentan en la Tabla 2.

Características	DIMEx100 niños
Sujetos	50 niños y 50 niñas
Edad	De 10 a 14 años
Lugar de origen	Ciudad de México
Nivel de escolaridad	Primaria
Número de oraciones	5000
Longitud de las oraciones	De 7 a 15 palabras
Tipo de habla	Leída
Tipo de transcripción	Ortográfica

Tabla 2: Características generales del corpus DIMEx100

Con este corpus se crearon los modelos acústicos del módulo de reconocimiento (ver subsección 3.3). Pruebas preliminares mostraron que el reconocedor tenía un desempeño de 47.5% de razón de error por palabra (WER, por sus siglas en inglés). Esta medida cuantifica los errores cometidos a nivel palabra, entre más bajo es mejor. Este resultado es comparable con el WER 48.3% obtenido para la versión de adultos del corpus. En conjunto con un modelo del lenguaje para la tarea, esta medida de error debe disminuir. Además cabe recordar que el modelo de diálogo y el módulo de intérprete de lenguaje se diseñaron teniendo en mente que este tipo de errores pueden ocurrir.

El corpus DIMEx100 niños se ha convertido para el grupo en un recurso invaluable. Además de permitirnos crear un buen reconocedor de voz para niños, este corpus se pueden responder diferentes preguntas de investigación. Por ejemplo, las diferencias entre modelos acústicos para adultos y niños. Parte de estas preguntas se estudiaron en trabajo [MOY11].

4.3 Corpus de interacciones

Como parte del plan de desarrollo continuo y de evaluación del proyecto, se recolectó permanentemente el corpus de interacciones de usuarios del sistema. Este corpus se integró con las grabaciones de los usuarios y con un registro de la historia de la interacción entre usuario y el sistema funcional. Las grabaciones se realizaron en varias etapas del proyecto y se seleccionaron las mejoras. Adicionalmente, las grabaciones fueron extendidas con una etiquetación manual, con el fin de identificar el tipo de lenguaje de los usuarios cuando interactúan directamente con el sistema, para mejorar los modelos de diálogo del reconocedor de voz y para enriquecer los modelos acústicos. En resumen se etiquetaron tres aspectos de cada interacción del usuario con el sistema:

- La transcripción ortográfica de lo dicho
- Los errores en el audio se identificaron y marcaron. Los errores más comunes fueron, que el sistema se escuchaba a sí mismo, que no escuchaba la oración completa del usuario o que otra persona, además del usuario, había hablado.
- La interpretación que correspondía a dicha interacción. Por ejemplo, la pregunta *¿es rojo?* se interpretó como *feature(red)*.

La Tabla 3 muestra las propiedades de este corpus; como vemos de un total de 172 usuarios que se grabaron interactuando con el sistema, se identificaron 1,765 elocuciones útiles para nuestros propósitos de mejorar, tanto los modelos acústicos como del lenguaje. No obstante, es importante notar que de estas elocuciones 569 son diferentes entre sí y el resto constituye repeticiones. Basado en este recurso se elaboró el trabajo [FAB11].

Característica	Corpus Interacciones
Número de interacciones	172
Total de elocuciones	1765
Tipos de elocuciones	569
Total de palabras	4212
Tipos de palabras	477
Duración de corpus	6h 44m

Tabla 3: Características del corpus de interacciones

5 Evaluación

La evaluación de la funcionalidad del sistema fue un objetivo establecido desde el inicio del proyecto y una tarea se llevó a cabo de manera constante. Para la evaluación de un sistema conversacional no existe a la fecha un consenso sobre una metodología única de evaluación. Por una parte, hay quienes se manifiestan por evaluar los componentes del sistema por separado (i.e., módulos) de manera independiente, mientras que otros insisten en que los sistemas deben evaluarse vistos en su totalidad. La primera opción tiene la ventaja de que para las tareas de los módulos existen pruebas ya definidas que se pueden automatizar; sin embargo, es difícil garantizar que una vez unidos los módulos el sistema funcione correctamente. Por otro lado, la evaluación de un sistema conversacional en su totalidad dificulta la automatización del levantamiento de información, ya que este tipo de evaluaciones se basa en la percepción que del sistema tienen sus usuarios.

Independientemente del tema de la automatización, también existen los investigadores dan mayor valor a las evaluaciones de tipo cuantitativo, mientras que para otros no existe una mejor forma de conocer la utilidad y la percepción de un sistema que la que se puede recoger de la opinión de los usuarios, es decir de manera cualitativa. Para conocer el desempeño del sistema Adivina la Carta revisamos distintas metodologías y decidimos trabajar bajo el marco teórico de PARADISE que toma en cuenta tanto los aspectos cuantitativos como cualitativos para una evaluación integral de las aplicaciones [WALKER97].

Conforme el sistema fue evolucionado, pero sobre todo, a partir de su puesta en operación en el museo Universum se han realizado diferentes evaluaciones. En la subsección 5.1 se presentan los resultados de una primera evaluación, realizada con motivo de la apertura del módulo. Esta evaluación sólo considera aspectos cualitativos. En la subsección 5.2 se presentan los resultados de la segunda evaluación. Esta evaluación se realizó con el módulo en operación y después de una actualización mayor al sistema. Finalmente, la subsección 5.3 muestra los resultados de la evaluación completa tomando en cuenta tanto aspectos cuantitativos como cualitativos bajo el esquema mencionado de PARADISE.

5.1 Primera evaluación

La primera evaluación del sistema se hizo sobre la primera versión del sistema en diciembre de 2009, con motivo de la puesta en operación del módulo en Universum. Para esta evaluación utilizamos el cuestionario sobre elementos cualitativos del marco teórico PARADISE, adaptado para que los conceptos fueran comprensibles para los niños. Este cuestionario tiene como fin conocer la percepción de los usuarios del sistema a través de ocho aspectos que resumen el desempeño del sistema: el entendimiento de la voz, el reconocimiento de la voz, la facilidad de la tarea, el ritmo de la interacción, la experiencia necesaria para la interacción, el tiempo de respuesta del sistema, la comportamiento esperado y su uso futuro. El cuestionario se muestra en la Tabla 4.

1. ¿Fue fácil de entenderle al sistema?
2. ¿Te entendió el sistema lo que dijiste?
3. ¿Fue fácil de adivinar la carta?
4. ¿Fue rápido el juego?
5. ¿Sabías que contestarle a Golem?
6. ¿El sistema se tardaba en contestarte?
7. ¿El sistema funcionó como esperabas?
8. ¿Volverías a jugar con Golem?

Tabla 4: Cuestionario preliminar de satisfacción de usuario basado en PARADISE.

Para la evaluación pedimos a 10 niños con edades entre 10 y 14 años interactuar con el sistema. Los niños no tenían experiencia previa con el sistema. De estas interacciones observamos que todos los niños concluyeron la tarea, que consistía en terminar el juego, que utilizaron en promedio 33.4 turnos y sólo cuatro de ellos lograron adivinar la carta. Una vez terminado el juego, cada niño contestó un cuestionario. La Tabla 5 resume los resultados obtenidos. De estos resultados encontramos, que los niños perciben el juego como una tarea difícil, esto se ve en los aspectos de la facilidad de la tarea y la experiencia necesaria para resolverla; estos dos aspectos fueron calificados con un bajo desempeño. También, cabe destacar que los usuarios percibieron que el sistema no los entendía, esto es común en los sistemas conversacionales, ya que el reconocedor de voz constituye en muchas ocasiones un cuello de botella para un buen desempeño del sistema.

Aspecto	Porcentaje
Entendimiento de voz	90.00%
Reconocimiento de voz	50.00%
Facilidad de tarea	50.00%
Ritmo de interacción	80.00%
Experiencia necesaria	50.00%
Tiempo de respuesta	70.00%
Comportamiento esperado	60.00%
Uso futuro	90.00%

Tabla 5: Porcentaje de respuestas "si" a las preguntas del cuestionario de satisfacción de usuario

5.2 Segunda evaluación

Esta evaluación se llevó a cabo en junio del 2010 para valorar el impacto de una actualización mayor realizada al sistema. Como parte de esa actualización se reestructuró la tarea, para quedar tal y como se presentó en la subsección 3.2, además de que se agregaron estrategias de recuperación. El escenario y las condiciones de evaluación fueron los mismos de la primera evaluación: solicitamos a 10 niños con edades entre 10 y 14 años interactuar con el sistema, los niños no tenían experiencia previa con el sistema, y al finalizar les aplicamos el cuestionario de satisfacción del usuario. En esta ocasión se observó una disminución en el número de turnos, de 33.6 de la versión anterior a 15.5. Consideramos, que en esta ocasión las mejoras al sistema contribuyeron a hacer los diálogos más cortos y la interacción más ágil, propiedades deseables en un sistema conversacional.

Aspecto	Porcentaje
Entendimiento de voz	90.00%
Reconocimiento de voz	50.00%
Facilidad de tarea	30.00%
Ritmo de interacción	90.00%
Experiencia necesaria	70.00%
Tiempo de respuesta	50.00%
Comportamiento esperado	60.00%
Uso futuro	100.00%

Tabla 6: Porcentaje de respuestas "si" a las preguntas del cuestionario de satisfacción de usuario

La Tabla 6 resume los resultados para esta evaluación. Vemos que la percepción sobre la facilidad de la tarea bajó y después de analizar nuestra experiencia de ver a los niños jugando con el sistema, hemos encontrado que para los niños resulta difícil elaborar las preguntas. No obstante, vemos que los niños percibieron mejor el ritmo de la interacción, mejora que creemos que tiene una causa en la mejora del modelo de diálogo. La percepción del tiempo de respuesta también bajó, pero esto se lo atribuimos que a través de las estrategias de recuperación ahora se hace al niño consciente de cuando el sistema no entendió o no escucho.

Finalmente para corroborar los resultados que evidenciaran la mejora del sistema con respecto a la versión anterior, comparamos los tiempos mínimos y máximos de duración de los diálogos de esta versión del sistema con respecto a los de la versión anterior. La Tabla 7 resume estos resultados. En la tabla se puede apreciar que aunque los promedios son similares para el tiempo y no lo son para el número de elocuciones: la mayor diferencia se encuentra en el rango en que los diálogos ocurren; mientras que con la primera versión del sistema tuvimos diálogos o muy cortos o muy largos, en la segunda versión la duración es más estándar. Esto confirma que los modelos de diálogos trabajados y las estrategias de recuperación mejoraron el comportamiento del sistema.

Factor	Versión anterior	Versión evaluación
Duración promedio	4m 41s	4m 49s
Máxima duración	8m 13s	6m 19s
Mínima duración	0m 37s	3m 23s
Elocuciones promedio	21	17
Número máximo de elocuciones	37	14
Número mínimo de elocuciones	2	12

Tabla 7: Comparación de duración sistema con versión anterior

5.3 Evaluación integral

Esta tercera evaluación tuvo el objetivo de medir el desempeño del sistema de manera más exhaustiva. La evaluación buscaba además de evaluar la efectividad en el intercambio de información entre el usuario y el sistema, si los usuarios eran capaces de terminar la tarea y conocer su nivel de satisfacción. Integramos las medidas de eficiencia y eficacia previstas en el marco teórico de PARADISE, pero aprovechando nuestra experiencia en las evaluaciones anteriores, hicimos algunas modificaciones, en particular al cuestionario de satisfacción del usuario. Un cambio consistió en ampliar y ajustar las preguntas con un lenguaje más directo para el niño y para dejar más claros conceptos como “dificultad”. Un segundo cambio fue que en lugar de tomar como respuestas únicamente “sí” o “no” utilizamos una escala Likert que dio más opciones de respuesta al niño, pero que nos siguió permitiendo hacer un análisis de respuestas positivas o negativas para comparar los resultados con los obtenidos en las evaluaciones anteriores. La Tabla 8 presenta la versión final de las preguntas.

Para esta evaluación pedimos a 30 niños con edades entre los 10 y 16 años y sin experiencia previa, interactuar con el sistema. De los 30 niños, 15 fueron niños y 15 niñas. En esta ocasión además de guardar los registros de las interacciones y de aplicar los cuestionarios grabamos audio y videos de las interacciones, los cuales fueron analizados posteriormente.

1. ¿Entendiste lo que Golem decía?
2. ¿Golem entendió lo que tú decías?
3. ¿Sábías que preguntarle a Golem?
4. ¿Tuviste tiempo suficiente para pensar las preguntas?
5. ¿Te gustó el ritmo del juego?
6. ¿Te gustó la velocidad de respuesta de Golem?
7. ¿El sistema funcionó como te lo imaginabas?
8. ¿Volverías a jugar con Golem?

Tabla 8: Cuestionario de satisfacción de usuario basado en PARADISE.

Aspecto	Porcentaje
Entendimiento de voz	93.00%
Reconocimiento de voz	40.00%
Facilidad de tarea	70.00%
Ritmo de interacción	96.00%
Experiencia necesaria	63.00%
Tiempo de respuesta	62.00%
Comportamiento esperado	93.00%
Uso futuro	92.00%

Tabla 9: Resultados de cuestionario de satisfacción de usuario

Aspecto	Cantidad
Promedio duración diálogos	4m 53s por diálogo
Promedio elocuciones usuario por diálogo	16.7 por diálogo
Promedio elocuciones sistema por diálogo	41.9 por diálogo
Juegos terminados	30
Usuarios que adivinaron	9
Promedio de preguntas hechas	3.77 por diálogo
Radio de error por palabra (WER)	59.00%
Promedio errores <i>no escucho</i>	1.0 por diálogo
Promedio errores <i>no entendió</i>	4.5 por dialogo

Tabla 10: Mediciones cuantitativas de las interacciones evaluadas

Los resultados de esta evaluación muestran que el sistema está cumpliendo con su cometido, es capaz de interactuar con los niños y todos los usuarios fueron capaces de completar la tarea. Sin embargo, los niños son jueces estrictos y perciben que el sistema no los entiende y que la tarea es muy difícil para ellos. En el primer caso, notamos que los valores cuantitativos corroboran esta percepción, donde notamos que existe un 59% de error por palabra y un promedio de 4.5 errores que involucran no entender lo que dijo el usuario. Sin embargo, es destacable que aún con esta carga de errores debido al reconocedor, el sistema terminó las interacciones. En el caso de la dificultad de la tarea, se corrobora, por el porcentaje de usuarios que adivinaron la carta. Como se discutió en la evaluación anterior, la formulación de la pregunta que el niño hace al sistema presenta una dificultad mayor que no habíamos previsto. Un resumen exhaustivo de la evaluación se puede encontrar en [VENEGAS11].

6 Conclusiones y resultados

Este documento presenta los resultados del proyecto Golem-Universum en el desarrollo del sistema Adivina la Carta.

1. **Implementar un sistema conversacional para que los visitantes del museo se acerquen a la Inteligencia Artificial.**

Mediante la instalación funcional del sistema Adivina la Carta en un módulo permanente en el museo de ciencias Universum se cumple el objetivo principal de este proyecto que es el de acercar a los visitantes del museo los últimos avances de la Inteligencia Artificial, en particular, a través de un sistema conversacional.

2. **Incrementar la robustez de los sistemas utilizados por el Grupo Golem.**

El desarrollo del sistema Adivina la Carta permitió al grupo Golem mejorar los sistemas que lo integran; se lograron:

- Avances en el manejador de diálogo, el reconocedor de voz, el intérprete del lenguaje y el visual.
- Control del ambiente acústico de la cabina que derivó en mejoras a los sistemas.

3. **Generación de productos de investigación como la presentación de artículos en congresos internacionales.**

- Las mejoras realizadas al sistema se reportaron en artículos publicados en revistas y congresos internacionales.
- Varios alumnos asociados al grupo Golem desarrollan sus tesis de licenciatura y maestría en aspectos específicos relacionados con distintos módulos del sistema, por ejemplo, el reconocimiento de voz, el entendimiento del lenguaje y los modelos del lenguaje. En la sección 6.1. se presenta una lista de los principales trabajos de investigación concluidos.

4. **Generar recursos humanos relacionados con el desarrollo de sistemas conversacionales**

Además de los alumnos asociados al grupo Golem que han desarrollado sus tesis de licenciatura y maestría en aspectos relacionados con el sistema, hemos integrado a estudiantes que realizan estancias temporales, así como a alumnos del programa Jóvenes a la Investigación. En la sección 6.3 agradecemos a los estudiantes que han realizado estancias dentro del proyecto Golem en Universum.

En la realización de las acciones específicas obtuvimos otros resultados:

a) **Desarrollo de módulos para integrar el sistema Adivina la carta.**

En la sección 3.3. se describen los módulos desarrollados y su funcionalidad para el correcto funcionamiento del

b) **Modelación de la tarea.**

Para modelar la tarea se decidió utilizar la técnica de Mago de Oz, la cual permitió recopilar un corpus de experimentos para realizar esta tarea y para integrar varios recursos que contribuyeron al desarrollo del sistema. La recopilación se realizó con ayuda del personal del museo Universum. Esta tarea se describe con mayor amplitud en el capítulo 4.1 de este reporte.

c) **Integración del corpus *DIMEx100 niños* y creación de un reconocedor para la voz infantil en español de México.**

El capítulo 4.2. de este reporte describe el proceso de recolección e integración del corpus, así como los beneficios que se obtuvieron para el reconocimiento de voz.

d) **Creación de un registro de interacciones con usuarios reales del sistema, en sus diferentes versiones, el cual fue etiquetado por lingüistas del Grupo Golem.**

El registro permitió integrar un corpus con interacciones de usuarios reales con las diversas versiones del sistema. El registro se extendió con la etiquetación manual realizada por los lingüistas del Grupo Golem. Los detalles de esta actividad pueden consultarse en la sección 4.3.

e) **Evaluación de la percepción de los usuarios del sistema *Adivina la Carta*.**

Una tarea constante del proyecto fue la evaluación del sistema. Durante el desarrollo se hicieron tres evaluaciones, centradas en aspectos cualitativos y una integral en la que se tomaron en cuenta tanto aspectos cualitativos como cuantitativos. De estas evaluaciones destacamos dos resultados; por una parte, que se requieren esfuerzos de investigación para mejorar el reconocimiento de voz y, en consecuencia, la percepción que los niños tienen sobre el desempeño del sistema. Por otra, los resultados demuestran la mejora del sistema usando una estructura más modular y estrategias de recuperación y, aunque la percepción y el desempeño real del reconocedor reste calidad a la interacción, los visitantes del museo pueden interactuar y jugar con el sistema. El capítulo 5 de este reporte se dedica a describir el proceso de evaluación y el detalle de los resultados.

6.1 Productos de investigación

La siguiente es la lista de artículos publicados, tanto en revistas como en congresos internacionales, en los que se reportan aspectos científicos del desarrollo del sistema Golem.

- Meza, I., Pérez-Pavón, P., Salinas, L., Avilés, H., Pineda, L. A.: A Multimodal Dialogue System for Playing the Game Guess the Card. *Procesamiento de Lenguaje Natural*. España. 44:131-138. (2010).
- Meza, I., Salinas, L., Venegas, E., Castellanos, H., Chavarria, A., Pineda, L.A.: Specification and Evaluation of a Spanish Conversational System Using Dialogue Models. En: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010. LNCS (LNAI), vol. 6433, pp. 346–355. 2010.
- Pineda, L.A., Meza, I., Salinas, L.: Dialogue Model Specification and Interpretation for Intelligent Multimodal HCI. In: Kuri-Morales, A., Simari, G. (eds.) IBERAMIA 2010.

LNCS (LNAI), vol. 6433, pp. 20–29. 2010 .

- Tapia, G., Meza, I., Pineda, L. A. Language Models for Name Recognition in Spanish Spoken Dialogue Systems. *Research in Computer Science*. 2010.
- Venegas, E., Meza, I. and Pineda, L.: Evaluation of a Dialogue System for Children based on an Interaction-Oriented Cognitive Architecture. *Procesamiento de Lenguaje Natural*, Vol. 46, pp. 113-110, 2011.
- Fabian, A., Hernández, M., Pineda, L. and Meza, I.: Contextual Semantic Processing for a Spanish Dialogue System Using Markov Logic. *Proceedings MICAI 2011*. LNCS (LNAI), Vol. 7094, pp. 258-266, 2011.
- Moya, E., Hernández, M., Pineda, L. and Meza, I.: Speech Recognition with Limited Resources for Children and Adult Speakers. *Tenth Mexican International Conference on Artificial Intelligence - Special Session - Revised Papers*, Vol. 2276, pp. 57-65, 2011.

Durante este proyecto se llevaron a cabo las siguientes tesis de licenciatura

- Esther Venegas, (Supervisor, Pineda, L.): Evaluación del sistema conversacional con español hablado “Golem–Adivinala Carta” en el museo Universum. Tesis de licenciatura FFyL-UNAM. vol --, pp 91. 2011.
- Edith Moya, (Supervisor, Meza, I. y Hernández, M.): “Entrenamiento dinámico de modelos acústicos de reconocedores de voz para los corpora en español de México: DIMEx100 niños y adultos Tesis de Licenciatura: UTM. vol --, pp 115. 2011.
- Aldo Fabian, (Supervisor, Meza, I. y Hernández, M.): “Desarrollo e implementación de un párser semántico para el módulo Golem-UNIVERSUM” Tesis de licenciatura UTM. vol --, pp 126. 2011.

Como parte de las demostraciones se elaboraron los siguientes videos demostración:

- Guess the card system, <http://www.youtube.com/watch?v=VSEZGaabSHs>
- Previous Version of Guess the Card System, http://www.youtube.com/watch?v=lg7q5w_kQIE
- Guess the card system, <http://www.youtube.com/watch?v=YyDqqOQczo>

Además alumnos del programa de Jóvenes a la investigación realizaron sus proyectos en relación al proyecto:

- Diseño de la Dinámica de Conversación para el proyecto Golem-Universum, por Carlos Enrique Hernandez Tenorio (Junio 2009). Tercer lugar en cartel.
- Evaluación del proyecto Mago de Oz, por Norma Verónica Trinidad Hernández (Junio

2010).

La siguiente es una lista de pláticas impartidas por los miembros del proyecto en los que se presentaron diferentes aspectos del proyecto.

- Meza, I.: “Un sistema conversacional y multimodal para una aplicación de museo”. Universidad Tecnológica de la Mixteca. Huajupan de León, Oaxaca, 8 de junio de 2010.
- Pineda, L.: “Especificación e Interpretación de Modelos de Diálogos para la Interacción Hombre-Máquina”. Universidad Politécnica de Sinaloa, Mazatlán, 7 de junio de 2010.
- Meza, I.: “Parseo e interpretación semántica a través de Métodos estadísticos”. Universidad Tecnológica de la Mixteca. Huajupan de León, Oaxaca, 8 de junio de 2010.
- Castellanos, H.: “Phonetic labeling of DIMEx100 adults and children corpora”. IIMAS, UNAM, 5o de Marzo de 2010. Dentro del taller del proyecto Golem.
- Fabian, A.: “Semantic parser for “guess the card” system”. IIMAS, UNAM, 5o de Marzo de 2010. Dentro del taller del proyecto Golem.
- Moya, E.: “Evaluation of speech recognizers using DIMEx100 adults and children corpora”. IIMAS, UNAM, 5o de Marzo de 2010. Dentro del taller del proyecto Golem.
- Venegas, E.: “Evaluation of “guess the card” system with the PARADISE framework”. IIMAS, UNAM, 2o de Marzo de 2010. Dentro del taller del proyecto Golem.
- Méza, I.: “Guess the card” project”. IIMAS, UNAM, 1o de Marzo de 2010. Dentro del taller del proyecto Golem. 1o de Marzo de 2010. Dentro del taller del proyecto Golem.
- Pineda, L.: “Arquitectura Cognitiva e Inferencia Perceptual”. Seminario SIAV, Grupo de señales, imágenes y ambientes virtuales. Posgrado CIC-UNAM, 5 de febrero de 2010.
- Pineda, L.: “El proyecto Adivina la Carta: Golem en Universum”, Museo Universum. 8 de septiembre de 2010.
- Pineda, L.: Panelista en “Tecnologías habilitadores” en el Encuentro Nacional sobre Envejecimiento y Salud. Instituto de Geriatria. Secretaría de Salud. 26 de agosto de 2010.
- Meza, I.: Specification and Evaluation of a Spanish Conversational System Using Dialogue Models. Conferencia IBERAMIA 2010, Bahía Blanca, Argentina. 3 de noviembre de 2010.
- Pineda, L.: Specification and Evaluation of a Spanish Conversational System Using Dialogue Models. Taller de Inteligencia Artificial, Posgrado de Ciencia e Ingeniería de la Computación, IIMAS, UNAM. 3 de diciembre de 2010.
- Venegas, E.: Evaluación del sistema conversacional con español hablado “Golem-Adivina la carta” en el museo Universum. 5º coloquio de lingüística computacional en la UNAM. Facultad de Filosofía y Letras. Agosto de 2011.
- Meza, I.: Entendimiento en el Sistema Conversacional Golem-Universum. 5º coloquio de lingüística computacional en la UNAM. Facultad de Filosofía y Letras, UNAM. Agosto de 2011.

6.2 Difusión en medios

El proyecto causó un interés de diferentes medios de comunicación. A continuación presentamos una lista de las entrevistas relacionadas al proyecto Golem-Universum.

- Pineda, L.: Participación en programa “Perfiles” (en vivo). Radio UNAM, 11 de enero de 2010.
- Meza, I.: Cápsula en Venga la Alegría sobre el proyecto Golem Universum. TV Azteca Canal 13. 11 de enero de 2010.
- Pineda, L.: Entrevista para el noticiero de México, Canal de la Presidencia, cable 412 y canal 22. 20 de enero de 2010.
- Pineda, L.: Entrevista telefónica con Josefina C. Herrera acerca de Golem en Universum, Radio Enfoque 100.1 FM y Radio Mil 1000 AM. Lunes 1o de febrero de 2010.
- Pineda, L.: Entrevista para noticieros de TV Azteca Canal 13 acerca de “Golem en Universum: Adivina la Carta”, Reportero Raymundo Rivaplacio, Museo Universum, 3 de febrero de 2010 a las 12:00. Transmisión: noticiero Canal 13 del 4 de febrero.
- Pineda, L.: Entrevista acerca del Proyecto Golem en el programa de radio “Venga a tomar café con nosotros”. Conductor Sergio de Régules. Transmitido 5 de febrero a las 20:05 Hrs. Radio UNAM 860 AM.
- Pineda, L.: Entrevista en vivo Grupo Imagen 90.5 FM radio, acerca de Inteligencia Artificial “Golem en Universum: Adivina la Carta”, programa “Imagen en la Ciencia”, entrevistador Rolando Isita. Domingo 7 de marzo de 2010.
- Avilés, H., Meza, I., Rascón, C.: Participación en el programa “Ingeniería en Marcha” radio UNAM. Entrevistadores: Ing. Ernesto Mendoza y Alejandra Torres, Martes 20 de julio de 2010.
- Pineda, L.: Reportaje sobre proyecto Golem en programa Reactor FTV canal 4. Conductor Leonardo Ferrera. Sábado 18 de septiembre de 2010.

A continuación se presenta una lista de las notas periodísticas sobre el proyecto Golem-Universum:

- Milenio TV: Universum su cumpleaños, nota en 13 de diciembre de 2009.
- La Jornada: En Universum es posible dialogar y jugar cartas con el robot Golem, nota 13 de diciembre de 2009.
- La Crónica: El Golem, robot que habla con los niños y les dota de conocimientos, nota el 14 de diciembre de 2009.
- Milenio: UNAM expone en Universum robot inteligente para niños, 25 de enero de 2010.
- Gaceta UNAM: Robótica para niños con Golem, en Universum, Reportero: Omar Páramo pp 8-9. 28 de enero de 2010:
- Revista ¿Cómo ves?: Reportaje “Golem: Un robot para platicar”, Año 12, No. 135,:

Reportero: “Guillermo Cárdenas”, pp 22-24, Febrero de 2010.

- La Crónica: Preparan la Expo Robótica 2010 con programas de divulgación, concursos y vinculación empresarial. 21 de Octubre de 2010.

La siguiente es una lista de eventos en los que se ha demostrado al sistema Adivina la Carta:

- Taller de Inteligencia Artificial del Posgrado de Ciencia e Ingeniería de la Computación. IIMAS, UNAM. Diciembre 2009.
- IV Congreso Nacional "Responsabilidad Social e Innovación en la Ingeniería", WTC, D.F. México. 3 al 5 de mayo.
- Expo Robótica, Expo Reforma, D.F. México. 4 al 7 de noviembre.

6.3 Agradecimientos

Agradecemos el apoyo del museo de ciencias Universum, en particular a:

René Drucker Colín, *Director General, DGDC*
Lourdes Guevara Saldaña, *Coordinadora de Museos*
Gabriela Sara Guzzy Arredondo, *Subdirectora de Universum*
Luis Alberto Morales Castillo, *Subdirector de Conservación*
Emmanuel Toscano Mendoza, *Jefe del Departamento de Diseño*
Germán Albizuri Caballero, *Apoyo Técnico*
Pablo Flores Herrera, *Asesor de sonido*
Esteban Estrada De Pavia, *Ingeniero de Sonido*
Esteban Monroy Audelo, *Apoyo Técnico*
Jimena Reyes Pimentel, *Jefa del Departamento de Comunicación Gráfica*
Ana Lara Velázquez, *Diseño Gráfico*
María de la Luz Oliva Agonizantes, *Diseño Gráfico*
Addina Cuerva Espinosa, *Jefa de la Unidad de Cómputo*
Claudia Hernández García, *Jefa del Área de Contenidos*
Brenda Aurora Flores Pérez, *Coordinadora del proyecto*

Al personal y alumnos asociados del IIMAS que participaron en diferentes etapas del proyecto:

Dr Fabian Garcia Nocetti, *Director del IIMAS*
M. en C. Wendi Elizabeth Aguilar Martínez, *Asesora técnica*
Ing Martín Fuentes Cruz, *Asesor técnico*
Ing Joel Durán Ortega, *Asesor técnico*
Ing Ricardo Villareal Martínez, *Asesor técnico*
Carlos Enrique Hernández Tenorio, *Jóvenes a la Investigación*
Norma Verónica Trinidad Hernández, *Jóvenes a la Investigación*
Edith Moya García, *estancia profesional y tesista UTM*
Aldo Ernesto Fabian Aguilar, *estancia profesional y tesista UTM*
Karen Soriano Rea, *estancia profesional de las UTM*
Nashielly Vasquez Bautista, *estancia profesional UTM*
Myriam Karenina Reyes Sánchez, *estancia profesional UTM*
Ing. Tania Zulema Pérez Núñez, *estancia profesional*

Además de los colaboradores externos:

Lucian Galescu, *Investigador Asociado*

León Soriano, *Ingeniero de sonido*

Diego Alvarez, *Ilustrador*

A Participantes

El grupo Golem que participó en el desarrollo del sistema Adivina la Carta está compuesto por:

- Dr. Luis Pineda, Lider de proyecto, filosofía del proyecto, modelos de diálogos y arquitectura cognitiva
- M.C. Paty Pérez Pavón, Supervisión 1ª versión, reconocimiento e interpretación de la tarea, modelos de diálogo
- Dr. Ivan V. Meza Ruiz, Reconocimiento e interpretación del lenguaje, administración de la tarea
- Ing. Lisset Salinas Pinacho, Programación de modelos de diálogo
- Lic. Esther Venegas Briones, Diseño y aplicación de evaluaciones
- Lic. Hayde Castellanos Vargas, Coordinación de sesiones y transcripción de registros
- Lic. Alejandra Chavarría , Coordinación de sesiones y transcripción de registros
- Dr. Héctor Avilés Arriaga, Sistema de visión
- Dr. Caleb Rascón Estebane, Sistema de audio

B Modelos de diálogo

A continuación se presentan la especificación de los modelos de diálogos que modelan el protocolo de interacción entre el sistema Adivina la Carta y los usuarios.

Modelo de diálogo principal: main.dm

```
% Copyright (c) 2009-      Luis Pineda (http://leibniz.iimas.unam.mx/~luis),
%                          Elia Patricia Pavon,
%                          Ivan V. Meza (http://turing.iimas.unam.mx/~ivanvladimir),
%                          Lisset Salinas Pinacho
diag_mod(main,
[
  % Situación inicial, despliega la pantalla inicial con 3 botones
  [
    id ==> is,
    type ==> neutral,
    out_pairs ==> [
      empty:despliega_botones => opcion_boton
    ] ],
  % Situación click, espera que se pulse uno de los tres botones, ya sea para iniciar el
  juego, ver el video o terminar la aplicación
  [
    id ==> opcion_boton,
    type ==> click,
    out_pairs ==> [
      juego:empty => ve_subdialogo_start,
      video:reproduce_video => is,
      cerrar:cerrar_app => fs
    ] ],
  % Situación recursiva, entra en un subdiálogo con las condiciones iniciales del sistema
  (pregunta de nombre, edad, si se quieren las instrucciones)
  [
    id ==> ve_subdialogo_start,
    type ==> recursive(start),
    out_pairs ==> [
      sigue(Carta):empty => ve_subdialogo_game(Carta),
      salida:termina => fs
    ] ],
  % Situación recursiva, entra en un subdiálogo donde se inicia la sesión de preguntas
  [
    id ==> ve_subdialogo_game(Model),
    type ==> recursive(game(Model)),
    out_pairs ==> [
      sigue(Carta):empty => ve_subdialogo_vision(Carta),
      salida:termina => fs
    ] ],
  % Situación recursiva, entra en un subdiálogo donde se evalúa si la carta mostrada al
  sistema es la que selecciono el mismo
  [
    id ==> ve_subdialogo_vision(Carta),
    type ==> recursive(vision(Carta)),
    out_pairs ==> [
      sigue:empty => ve_subdialogo_end,
      salida:termina => fs
    ] ],
  % Situación recursiva, entra en un subdiálogo donde se toma la fotografía del usuario y se
  despide el sistema
  [
    id ==> ve_subdialogo_end,
    type ==> recursive(end),
    out_pairs ==> [
      salida:empty => fs
    ] ],
  % Situación final, termina la interacción y regresa a la pantalla inicial
```

```

        [
            id ==> fs,
            type ==> final,
            out_pairs ==> [
                salida:empty => empty
            ] ],
%Actos retoricos
[
    [
        id ==> despliega_botones,
        type ==> entradda,
        rht_acts ==> [
            initSpeechRec,
            clean,
            putButton('juego', 'boton_verde', 200, 300),
            putButton('video', 'boton_rojo', 600, 300),
            putButton('cerrar', 'boton_cierra', 780, 580)
        ] ],
    [
        id ==> reproduce_video,
        type ==> video,
        rht_acts ==> [
            clean,
            despliega_video('intro', '00#00#10'),
            duerme(3)
        ] ],
    [
        id ==> termina,
        type ==> fin,
        rht_acts ==> [
            despedida
        ] ],
    [
        id ==> cerrar_app,
        type ==> despedida,
        rht_acts ==> [
            clean,
            cierra
        ]
    ] ] ).

```

Modelo de diálogo para la presentación e instrucciones: start.dm

```

% Copyright (c) 2009-      Luis Pineda (http://leibniz.iimas.unam.mx/~luis),
%                          Elia Patricia Pavon,
%                          Ivan V. Meza (http://turing.iimas.unam.mx/~ivanvladimir),
%                          Lisset Salinas Pinacho
diag_mod(start,
[
    % Situación inicial, da un saludo
    [
        id ==> is,
        type ==> neutral,
        out_pairs ==> [
            empty:saludo => ve_subdialogo_name
        ] ],
    % Situación recursiva, entra en un subdiálogo donde se pregunta el nombre del usuario
    [
        id ==> ve_subdialogo_name,
        type ==> recursive(name),
        out_pairs ==> [
            sigue:empty => ve_subdialogo_age,
            salida:empty => fs
        ] ],
    % Situación recursiva, entra en un subdiálogo donde se pregunta la edad del usuario
    [
        id ==> ve_subdialogo_age,
        type ==> recursive(age),
        out_pairs ==> [
            sigue:empty => ve_subdialogo_explain,

```



```

ID:(_,ls1,nada:_=>_),
1,fs,ls1])

],
out_pairs ==> [
    nombre(N):di_nombre(N) => n1
] ],
% Situación para decir la cortesia al saludar
[
    id ==> n1,
    type ==> neutral,
    out_pairs ==> [
        empty:gusto_conocerete => fe
    ]],
% Situación final con funcionamiento correcto, se continua con la interacción
[
    id ==> fe,
    type ==> final,
    out_pairs ==> [
        sigue:empty => empty
    ] ],
% Situación final para terminar el juego
[
    id ==> fs,
    type ==> final,
    out_pairs ==> [
        salida:empty => empty
    ] ] ],

%Actos retoricos
[
    [
        id ==> pide_nombre,
        type ==> juego,
        rht_acts ==> [
            pide_nombre
        ] ],
    [
        id ==> di_nombre(Nombre),
        type ==> introduccion,
        rht_acts ==> [
            despliega('hola',''),
            hola(Nombre)
        ] ],
    [
        id ==> gusto_conocerete,
        type ==> saludo,
        rht_acts ==> [
            gusto
        ]],
    [
        id ==> no_entiendo,
        type ==> error,
        rht_acts ==> [
            despliega('hola',''),
            hola2
        ] ],
    [
        id ==> puedes_repetir,
        type ==> error,
        rht_acts ==> [
            error_nom1
        ] ],
    [
        id ==> no_escucho,
        type ==> error,
        rht_acts ==> [
            error_nom4
        ] ],
    [
        id ==> sin_sonido,

```



```

        1,fs,ls2(X)])
    ],
    out_pairs ==> [
        ok:         apply(when, [X>9, puede_jugar, no_puede_jugar, []]) =>
                    apply(when, [X>9, fe, fs, []]),
        no:no_confirmada => ls1
    ] ],
% Situación final en caso de que haya usuario y este se encuentre en el rango de edad
[
    id ==> fe,
    type ==> final,
    out_pairs ==> [
        sigue:empty => empty
    ] ],
% Situación final Estados de error y las respuestas que dará el sistema en caso de no
entender lo que dice el usuario o no escuchar
[
    id ==> fs,
    type ==> final,
    out_pairs ==> [
        salida:empty => empty
    ] ] ],
%Actos retoricos
[
    [
        id ==> pide_edad,
        type ==> juego,
        rht_acts ==> [
            pide_edad
        ] ],
    [
        id ==> confirmar_edad(Edad),
        type ==> revisa,
        rht_acts ==> [
            confirma(Edad)
        ] ],
    [
        id ==> puedes_repetir,
        type ==> error,
        rht_acts ==> [
            error_age1
        ] ],
    [
        id ==> puedes_repetir_con,
        type ==> error,
        rht_acts ==> [
            error_conf2
        ] ],
    [
        id ==> no_escucho,
        type ==> error,
        rht_acts ==> [
            no_escuche
        ] ],
    [
        id ==> sin_sonido,
        type ==> error,
        rht_acts ==> [
            error_nom3
        ] ],
    [
        id ==> no_confirmada,
        type ==> edad,
        rht_acts ==> [
            sin_carta,
            pide_edad
        ] ],
    [
        id ==> puede_jugar,
        type ==> introduccion,
        rht_acts ==> [

```

```

                despliega('puede_jugar',''),
                puede_jugar
            ] ],
            [
                id ==> no_puede_jugar,
                type ==> introduccion,
                rht_acts ==> [
                    despliega('no_puede_jugar',''),
                    no_puede_jugar
                ] ],
            [
                id ==> error(Concept),
                type ==> correction,
                rht_acts ==> [
                    error(Concept)
                ] ] ] ).

```

Modelo de diálogo para las instrucciones: explain.dm

```

% Copyright (c) 2009- Luis Pineda (http://leibniz.iimas.unam.mx/~luis),
%                      Elia Patricia Pavon,
%                      Ivan V. Meza (http://turing.iimas.unam.mx/~ivanvladimir),
%                      Lisset Salinas Pinacho
diag_mod(explain,
[
    % Situación inicial, pregunta al usuario si quiere escuchar las instrucciones
    [
        id ==> is,
        type ==> neutral,
        out_pairs ==> [
            empty:pregunta_quiere_instrucciones => ls1
        ] ],
    % Situación para escuchar la afirmación o negación para escuchar instrucciones, pregunta
    si se entendieron las mismas
    [
        id ==> ls1,
        type ==> listening,
        error_pairs ==> [
            noEntendi: apply(check_size_pattern,[
                ID:(_,ls1,noEntendi:_=>_),
                1,dice_instrucciones,puedes_repetir]) =>
                apply(check_size_pattern,[
                    ID:(_,ls1,noEntendi:_=>_),
                    1,ls2,ls1]),
            nada: apply(check_size_pattern,[
                ID:(_,ls1,nada:_=>_),
                1,no_escucho,sin_sonido]) =>
                apply(check_size_pattern,[
                    ID:(_,ls1,nada:_=>_),
                    1,fs,ls1])
        ],
        out_pairs ==> [
            ok:dice_instrucciones => ls2,
            no:dice_comienza_juego(confirma_empieza) => n1
        ] ],
    % Situación para escuchar la afirmación o negación para el entendimiento de las
    instrucciones, evalua la repetición de las instrucciones
    [
        id ==> ls2,
        type ==> listening,
        error_pairs ==> [
            noEntendi: apply(check_size_pattern,[
                ID:(_,ls2,noEntendi:_=>_),

```

```

1,empty,puedes_repetir]) =>

        apply(check_size_pattern,[
ID:(_,ls2,noEntendi:_=>_),
1,n1,ls2]),

nada:        apply(check_size_pattern,[
ID:(_,ls2,nada:_=>_),
1,no_escucho,sin_sonido]) =>

        apply(check_size_pattern,[
ID:(_,ls2,nada:_=>_),
1,fs,ls2])
],
out_pairs ==> [
ok:dice_comienza_juego(confirma_empieza) => n1,
no:        apply(check_size_pattern,[
ID:(_,ls2,no:_=>_),
1,empty,dice_instrucciones]) =>

        apply(check_size_pattern,[
ID:(_,ls2,no:_=>_),
1,n1,ls2])
] ],
% Situación para seleccionar la carta
[
id ==> n1,
type ==> neutral,
out_pairs ==> [
empty:empty => apply(get_random_topic,[fe])
] ],
% Situación final con funcionamiento correcto, se iniciará el juego
[
id ==> fe(Carta),
type ==> final,
out_pairs ==> [
sigue(Carta):empty => empty
] ],
% Situación final para terminar el juego
[
id ==> fs,
type ==> final,
out_pairs ==> [
salida:empty => empty
] ] ],

%Actos retoricos
[
[
id ==> pregunta_quiere_instrucciones,
type ==> juego,
rht_acts ==> [
opcion_s_n('que te explique','las reglas del juego')
] ],
[
id ==> dice_comienza_juego(Valor),
type ==> confirmacion,
rht_acts ==> [
caption(Valor)
] ],
[
id ==> puedes_repetir,
type ==> error,
rht_acts ==> [
error_conf2
] ],
[
id ==> no_escucho,
type ==> error,
rht_acts ==> [
error_nom4
] ],

```

```
] ],  
[  
    id ==> sin_sonido,  
    type ==> error,  
    rht_acts ==> [  
        no_escuche  
    ] ],  
[  
    id ==> dice_instrucciones,  
    type ==> explicacion,  
    rht_acts ==> [  
        despliega('bienvenida', ''),  
        explica,  
        despliega('cartas', ''),  
        explica2,  
        explica3,  
        despliega('color', ''),  
        explica4,  
        despliega('forma', ''),  
        explica5,  
        despliega('se_parece', ''),  
        explica6,  
        despliega('ejemplo_preguntas', ''),  
        explica7,  
        explica8,  
        pregunta_entiendes  
    ] ] ] ).
```



```

3, disculpa, puedes_repetir, [20, 21, 22, 23, 24])) =>

    apply(check_size_pattern, [
        ID:(_, ls2(Y), noEntendi:_=>_),
        3, ls1, ls2(Y)]),

        nada:    apply(error_message, [
            ID:(_, ls2(Y), nada:_=>_),
            2, no_escuche, puedes_repetir, [5, 6, 7, 8, 9]]) =>

            apply(check_size_pattern, [
                ID:(_, ls2(Y), nada:_=>_),
                2, fs, ls2(Y)])

    ],
    out_pairs ==> [
        ok: escucho_carta(Y) => ls3(Y),
        no: disculpa => ls1
    ] ],
    % Situación para escuchar, ya que el usuario a confirmado que adivino la carta, se
    verifica que se escucho la carta correcta
    [
        id ==> ls3(Y),
        type ==> listening,
        error_pairs ==> [
            noEntendi:    apply(error_message, [
                ID:(_, ls3(Y), noEntendi:_=>_),
                3, empty, puedes_repetir, [20, 21, 22, 23, 24]]) =>

                apply(check_size_pattern, [
                    ID:(_, ls3(Y), noEntendi:_=>_),
                    3, fe(Carta), ls3(Y)]),

            nada:    apply(error_message, [
                ID:(_, ls3(Y), nada:_=>_),
                2, no_escuche, puedes_repetir, [5, 6, 7, 8, 9]]) =>

                apply(check_size_pattern, [
                    ID:(_, ls3(Y), nada:_=>_),
                    2, fs, ls3(Y)])

        ],
        out_pairs ==> [
            ok: con_carta => fe(Carta),
            no: una_oportunidad => n1
        ] ],
    % Situación final con funcionamiento correcto, se continua con el juego
    [
        id ==> fe(Carta),
        type ==> final,
        out_pairs ==> [
            sigue(Carta): empty => empty
        ] ],
    % Situación final para terminar el juego
    [
        id ==> fs,
        type ==> final,
        out_pairs ==> [
            salida: empty => empty
        ] ] ],
    %Actos retóricos.
    [
        [
            id ==> pide_primer_pregunta,
            type ==> peticion,
            rht_acts ==> [
                pide_pregunta
            ] ],
        [
            id ==> seguro,
            type ==> carta,
            rht_acts ==> [
                seguro_carta
            ]
        ]
    ]

```

```

] ],
[
    id ==> disculpa,
    type ==> carta,
    rht_acts ==> [
        lo_siento,
        otra_pregunta
    ] ],
[
    id ==> no_entiendo_car,
    type ==> error,
    rht_acts ==> [
        error_feat3
    ] ],
[
    id ==> puedes_repetir(Z),
    type ==> error,
    rht_acts ==> [
        caption(Z,t_error)
    ] ],
[
    id ==> no_entiendo,
    type ==> error,
    rht_acts ==> [
        error_feat5
    ] ],
[
    id ==> no_escucho,
    type ==> error,
    rht_acts ==> [
        no_escuche
    ] ],
[
    id ==> sin_sonido,
    type ==> error,
    rht_acts ==> [
        no_escucho
    ] ],
[
    id ==> no_car(Imagen,Carta),
    type ==> descripcion,
    rht_acts ==> [
        despliega(Imagen,''),
        caption(Imagen,no,Carta )
    ] ],
[
    id ==> si_car(Imagen,Carta),
    type ==> descripcion,
    rht_acts ==> [
        despliega(Imagen,''),
        caption(Imagen,ok,Carta )
    ] ],
[
    id ==> repetida,
    type ==> peticion,
    rht_acts ==> [
        repetida
    ] ],
[
    id ==> queda_otra,
    type ==> peticion,
    rht_acts ==> [
        otra_pregunta
    ] ],
[
    id ==> escucho_carta(Concepto),
    type ==> pregunta,
    rht_acts ==> [
        despliega('duda',''),
        caption(Concepto,pregunta)
    ] ],
] ],

```

```

[
    id ==> queda_una,
    type ==> peticion,
    rht_acts ==> [
        ultima_pregunta
    ] ],
[
    id ==> no_queda,
    type ==> peticion,
    rht_acts ==> [
        no_mas_preguntas
    ] ],
[
    id ==> una_oportunidad,
    type ==> confirmacion,
    pars ==> [],
    rht_acts ==> [
        una_oportunidad
    ] ],
[
    id ==> con_carta,
    type ==> carta,
    rht_acts ==> [
        con_carta
    ] ] ] ).

```

Modelo de diálogo para visualizar la carta: vision.dm

```

% Copyright (c) 2009- Luis Pineda (http://leibniz.iimas.unam.mx/~luis),
% Elia Patricia Pavon,
% Ivan V. Meza (http://turing.iimas.unam.mx/~ivanvladimir),
% Lisset Salinas Pinacho
diag_mod(vision(Carta),
[
    % Situación inicial, verifica si el usuario adivino la carta en el modelo de juego
    [
        id ==> is,
        type ==> neutral,
        out_pairs ==> [
            empty: apply(check_history4pattern, [
                game(Carta):(ls3(Y)=>ok:con_carta,final,::_=>_),
                1,con_carta(muestra),sin_carta(muestra))] =>
                apply(check_card, [
                    1,v2,v1])
            ] ],
        % Situación visual, cuando el usuario no adivino la carta de ante mano, muestra la carta
        vista por el sistema
        [
            id ==> v1,
            type ==> seeing,
            out_pairs ==> [
                carta(Carta_vista):muestra_sin(Carta_vista) => ls1(Carta_vista)
            ] ],
        % Situación visual, cuando el usuario adivino la carta de ante mano, muestra la carta
        vista por el sistema
        [
            id ==> v2(Y),
            type ==> seeing,
            out_pairs ==> [
                carta(Carta_vista):muestra_con(Carta_vista) => n1(Y,Carta_vista)
            ] ],
        % Situación escuchar, espera la afirmación de que la carta vista fue la carta mostrada
        % Compara la carta seleccionada por el usuario con la seleccionada por el sistema
        [
            id ==> ls1(Carta_vista),
            type ==> listening,
            error_pairs ==> [
                noEntendi: apply(error_message, [

```

```

ID:(_, ls1(Carta_vista), noEntendi: _=>_),
2, empty, puedes_repetir, [10, 11, 12, 13, 14])=>

apply(check_size_pattern, [
ID:(_, ls1(Carta_vista), noEntendi: _=>_),
2, n2(Carta_vista), ls1(Carta_vista)]),

nada: apply(error_message, [
ID:(_, ls1(Carta_vista), nada: _=>_),
1, no_escuche, puedes_repetir, [5, 6, 7, 8, 9]]) =>

apply(check_size_pattern, [
ID:(_, ls1(Carta_vista), nada: _=>_),
1, fs, ls1(Carta_vista)])
],
out_pairs ==> [
ok: apply(when2, [Carta, Carta_vista,
gano(Carta), perdio(Carta, Carta_vista), []]) =>
fe,
no:sin_carta(muestra) => v1
] ],
% Verifica que la carta seleccionada de ante mano por el usuario y la vista por el sistema
sean iguales
[
id ==> n1(Y, Carta_vista),
type ==> neutral,
out_pairs ==> [
empty: apply(when2, [Y, Carta_vista,
empty, diferentes, []]) =>

apply(when2, [Y, Carta_vista,
n2(Carta_vista), ls2(Y, Carta_vista), []])
] ],
% Cuando no entiende la afirmación del usuario verifica que la carta vista por el sistema
sea igual a la seleccionada por el mismo sistema
[
id ==> n2(Carta_vista),
type ==> neutral,
out_pairs ==> [
empty: apply(when2, [Carta, Carta_vista,
gano(Carta), perdio(Carta, Carta_vista), []]) =>
fe
] ],
% Situación escuchar, cuando la carta elegida de ante mano y la vista por el sistema son
diferentes
% espera la afirmación del usuario para continuar con la carta vista
[
id ==> ls2(Y, Carta_vista),
type ==> listening,
error_pairs ==> [
noEntendi: apply(check_size_pattern, [
ID:(_, ls2(Y, Carta_vista), noEntendi: _=>_),
2, no_entiendo, puedes_repetir, [15, 16, 17, 18, 19]]) =>

apply(check_size_pattern, [
ID:(_, ls2(Y, Carta_vista), noEntendi: _=>_),
2, n2(Carta_vista), ls2(Carta_vista)]),

nada: apply(check_size_pattern, [
ID:(_, ls2(Y, Carta_vista), nada: _=>_),
1, no_escuche, puedes_repetir, [0, 1, 2, 3, 4]]) =>

apply(check_size_pattern, [
ID:(_, ls2(Y, Carta_vista), nada: _=>_),
1, fs, ls2(Carta_vista)])
],
out_pairs ==> [
ok:no_elige_carta => n2(Carta_vista),
no:empty => n3(Y)
] ],
% Se ejecuta cuando el usuario prefiere continuar con la carta seleccionada de ante mano

```

```

[
  id ==> n3(Y),
  type ==> neutral,
  out_pairs ==> [
    empty:
      apply(when2,[Carta,Y,
        gano(Carta),perdio(Carta,Y),[]]) =>
        fe
  ] ],
% Situación final con funcionamiento correcto, continua la interacción
[
  id ==> fe,
  type ==> final,
  out_pairs ==> [
    sigue:empty => empty
  ] ],
% Situación final para terminar el juego
[
  id ==> fs,
  type ==> final,
  out_pairs ==> [
    salida:empty => empty
  ] ] ],
%Actos retoricos.
[
  [
    id ==> con_carta(Opt),
    type ==> peticion,
    rht_acts ==> [
      despliega(Opt,''),
      caption(Opt)
    ] ],
  [
    id ==> sin_carta(Opt),
    type ==> peticion,
    rht_acts ==> [
      despliega(Opt,''),
      caption(Opt)
    ] ],
  [
    id ==> muestra_sin(Carta_vista),
    type ==> check,
    rht_acts ==> [
      despliega(Carta_vista,''),
      caption(confirma_carta)
    ] ],
  [
    id ==> muestra_con(Carta_vista),
    type ==> check,
    rht_acts ==> [
      despliega(Carta_vista,''),
      caption(compara)
    ] ],
  [
    id ==> no_entiendo,
    type ==> error,
    rht_acts ==> [
      error_feat3
    ] ],
  [
    id ==> puedes_repetir(Z),
    type ==> error,
    rht_acts ==> [
      caption(Z,t_error)
    ] ],
  [
    id ==> no_escucho,
    type ==> error,
    rht_acts ==> [
      no_escucho
    ] ],
  [

```

```

        id ==> sin_sonido,
        type ==> error,
        rht_acts ==> [
            no_escuche
        ] ],
    [
        id ==> diferentes,
        type ==> carta,
        rht_acts ==> [
            cambio_opinion
        ] ],
    [
        id ==> gano(Carta),
        type ==> despedida,
        rht_acts ==> [
            despliega(Carta, ''),
            Carta,
            despliega('gana', '')
        ] ],
    [
        id ==> perdio(Carta, Carta_adevinada),
        type ==> despedida,
        rht_acts ==> [
            despliega(Carta_adevinada, ''),
            caption(Carta_adevinada, no),
            despliega(Carta, ''),
            era(Carta)
        ] ] ] ).

```

Modelo de diálogo para la despedida: end.dm

```

% Copyright (c) 2009-      Luis Pineda (http://leibniz.iimas.unam.mx/~luis),
%                          Elia Patricia Pavon,
%                          Ivan V. Meza (http://turing.iimas.unam.mx/~ivanvladimir),
%                          Lisset Salinas Pinacho
diag_mod(end,
[
    % Situación inicial, toma la fotografía del usuario
    [
        id ==> is,
        type ==> neutral,
        out_pairs ==> [
            empty:toma_foto => n1
        ] ],
    % Situación neutral, busca en la historia el nombre del usuario para despedirse
    [
        id ==> n1,
        type ==> neutral,
        out_pairs ==> [
            empty: apply(check_name, [1, foto_nombre, foto_nnombre]) => fs
        ] ],
    % Situación final, termina el juego
    [
        id ==> fs,
        type ==> final,
        out_pairs ==> [
            salida:empty => empty
        ] ] ],
%Actos retoricos
[
    [
        id ==> toma_foto,
        type ==> despedida,
        rht_acts ==> [
            tomaFoto,
            sonrie,
            cheese('listo')
        ] ],
    [

```

```
id ==> foto_nombre(Nombre),
type ==> despedida,
rht_acts ==> [
  clean,
  putImage('foto', 'foto', 0, 0),
  duerme(3),
  adios(Nombre)
] ],
[
  id ==> foto_nnombre,
  type ==> despedida,
  rht_acts ==> [
    clean,
    putImage('foto', 'foto', 0, 0),
    duerme(3),
    adios2
  ] ] ] ).
```

Referencia

- WEIZ66: Weizenbaum, Joseph, ELIZA - A Computer Program For the Study of Natural Language Communication, Communications of the ACM. 36-45. 1966.
- ALLEN06: Allen, J. F., N. Blaylock, et al., Chester: Towards a personal medical advisor, Biomedical informatics. 500-513, 2006.
- PINEDA09: Pineda, L.A., H. Castellanos, J. Cuétara, L. Galescu, J. Juárez, J. Llisterri, P. Pérez-Pavón, L. Villaseñor, The Corpus DIMEx100: Transcription and Evaluation, Language Resources and Evaluation. . 2009.
- AGUILAR09: Wendy Aguilar & Luis A. Pineda, Integrating Graph-Based Vision Perception to Spoken Conversation in Human-Robot Interaction, IWANN 2009, Part I, LNCS 5517. 789–796, 2009.
- PINEDA08: Pineda, L. A., Specification and Interpretation of Multimodal Dialogue Models for Human-Robot Interaction, Artificial Intelligence for Humans: Service Robots and Social Modeling. 3350, 2008.
- PINEDA10: Pineda, L., Meza, I., Salinas, L., Dialogue Model Specification and Interpretation for Intelligent Multimodal HCI, Proceedings IBERAMIA 2010. 20-29, 2010.
- MANN88: Mann, W.C., Thompson, Rhetorical Structure Theory: Towards a functional theory of text organization, Text. 243-281. 1988.
- LOWE04: Lowe, D., Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision. 91-110. 1994.
- MEZA10a: Meza, I., Pérez, P., Salinas, L., Aviles, H., Pineda, L., A Multimodal Dialogue System for Playing de Game Guess the Card, Procesamiento de Lenguaje Natural. 131-138. 2010.
- MEZA10b: Meza, I., Salinas, L., Venegas, E., Castellanos, H., Chavarría, A., Pineda, L., Specification and Evaluation of a Spanish Conversational System Using Dialogue Models, Proceedings of IBERAMIA 2010. 346-355, 2010.
- HUERTA99: Huerta, J.M., Chen, S.J., Stern, R.M., The 1998 Carnegie Mellon University Sphinx-3 Spanish Broadcast News Transcription System, Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. , 1999.
- DUTOIT96: Dutoit, T., Pagel, V., Pierret, N., Bataille, F., Van der Vreken, O., The Mbrola Project: Towards a set of high-quality Speech Synthesizers Free of User for Non-Commercial Purposes, Proceedings of ICSLP'96. 1393-1396, 1996.
- CHEYNER01: , The Open Agent Architecture, Journal of Autonomous Agents and Multi-Agent Systems. 143-148. 2001.
- HERNANDEZ09: Hernandez, T., Reporte: Diseño de la Dinámica de Conversación para el proyecto Golem-Universum, 2009
- MOY11: Edith Moya, E., Hernández, M., Pineda, L. and Meza, I., Speech Recognition with Limited Resources for Children and Adult Speakers, 2011
- FAB11: Aldo Fabian, Hernández, M., Pineda, L. and Meza, I., Contextual Semantic Processing for a Spanish Dialogue System Using Markov Logic, 2011
- WALKER97: Walker, M., Litman, A., Kamm, D.J., Abella, A., PARADISE: A framework for Evaluating Spoken Dialogue Agents., 271-280, 1997.
- VENEGAS11: Esther Briones, Meza, I. and Pineda, L., Evaluation of a Dialogue System for Children based on an Interaction-Oriented Cognitive Architecture, Procesamiento de Lenguaje Natural. 113-110. 2011.