

Benemérita Universidad Autónoma de Puebla  
Facultad de Ciencias de la Computación



Modelo computacional para la generación automática de  
diálogos de un dominio específico mediante el uso de  
técnicas de aprendizaje automático

T E S I S

Presentada para obtener el grado de:  
Doctor en Ingeniería del Lenguaje y del Conocimiento

P r e s e n t a:

*M. en C. José Andrés Vázquez Flores*

Director de Tesis:

*Dr. David Eduardo Pinto Avendaño*

Diciembre 2020



*Al recuerdo de mis padres Domi y Margarito,  
que estarán siempre presentes.*

# Agradecimientos

En primer lugar, agradezco a las instituciones el apoyo que me brindaron durante los estudios de doctorado.

A la Benemérita Universidad Autónoma de Puebla (BUAP), por la formación profesional que me brindó, misma que me ha permitido crecer como persona.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por la beca (No. 460682) recibida a través del Programa Nacional de Posgrados de Calidad (PNPC).

A la Vicerrectoría de Investigación y Estudios de Posgrado de la BUAP, por el apoyo económico brindado para la publicación del artículo JCR.

Al Departamento de Tecnologías de la Información de la Universidad Autónoma Metropolitana (UAM), Unidad Cuajimalpa por haberme recibido y permitido realizar la estancia de investigación, y al mismo tiempo a la Red Temática de Tecnologías del Lenguaje por el apoyo económico para la realización de dicha estancia.

Gracias al Posgrado en Ingeniería del Lenguaje y del Conocimiento de la Facultad de Ciencias de la Computación de la BUAP por permitirme llegar hasta aquí.

Agradezco a todas las personas que me han apoyado para la realización de este proyecto, de manera muy especial a mi asesor Dr. David Eduardo Pinto Avendaño, por ser el mejor de los guías en esta fantástica aventura llamada investigación, por su infinita paciencia, por su apoyo, comprensión y guía durante todo el doctorado, por la inmensa confianza que ha depositado en mí y sobre todo por tan generosa y maravillosa amistad desde hace tantos años.

Al Dr. Héctor Jiménez Salazar por el apoyo en la realización de la estancia de investigación en la UAM y por sus valiosos comentarios.

A los integrantes de mi Comité Tutorial, Dra. Darnes Vilariño Ayala, Dr. Arturo Olvera López, Dr. Juan Manuel González Calleros, Dr. Héctor Jiménez Salazar y al Dr. David Eduardo Pinto Avendaño, por el tiempo y dedicación que dieron a la revisión del presente trabajo de investigación, las recomendaciones, aportaciones y todo el apoyo que me han brindado.

A mi carnal Jesús Lavallo por todo el apoyo brindado y sobre todo por su amistad.

A mis compañeras y amigas del doctorado Bety y Yolanda con las que he vivido momentos entrañables.

A mis hermanos Gerardo, Lili, Chayito y Sergio por sus palabras de aliento en todo momento.

Gracias a mi esposa Flor e hijos Ari y Andy por estar siempre a mi lado y por su infinito cariño.

A todos muchas gracias.

# Índice general

<b>Agradecimientos</b>	<b>ii</b>
<b>Lista de figuras</b>	<b>vii</b>
<b>Lista de tablas</b>	<b>ix</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento de la investigación . . . . .	2
1.2. Objetivos de la investigación . . . . .	3
1.2.1. Objetivos específicos: . . . . .	3
1.3. Preguntas de investigación . . . . .	4
1.4. Aportaciones de la investigación . . . . .	4
1.5. Organización de la tesis . . . . .	5
<b>2. Marco Teórico</b>	<b>6</b>
2.1. Interacción humano-computadora . . . . .	6
2.2. Diálogo . . . . .	8
2.3. Sistemas de diálogo . . . . .	9
2.4. Entidades nombradas . . . . .	12
2.5. Corpus . . . . .	13
2.6. Técnica del Mago de Oz . . . . .	15
2.7. Aprendizaje automático . . . . .	16
2.7.1. Métodos de aprendizaje automático . . . . .	17
2.7.2. Tipos de aprendizaje automático . . . . .	17

2.8. Campo Aleatorio Condicional . . . . .	18
2.9. Inferencia Gramatical . . . . .	20
2.10. Ontología . . . . .	22
<b>3. Estado del Arte</b>	<b>26</b>
3.1. Modelos basados en reglas probabilísticas . . . . .	26
3.2. Modelos basados en procesos estocásticos . . . . .	28
3.3. Modelos basados en redes neuronales . . . . .	29
<b>4. Sistema de Diálogo propuesto</b>	<b>35</b>
4.1. Estructura general del Sistema de Diálogo . . . . .	35
4.2. Reconocimiento del Lenguaje . . . . .	37
4.2.1. Reconocedor de Entidades Nombradas (NER) . . . . .	37
4.2.1.1. Descripción general del entrenamiento de un modelo NER . . . . .	37
4.3. Inferencia Gramatical . . . . .	43
4.3.1. Conceptos básicos . . . . .	43
4.3.2. Modelo de inferencia gramatical propuesto . . . . .	45
4.4. Generación de la respuesta . . . . .	48
4.4.1. Administración de la base de conocimiento . . . . .	50
4.4.2. Verificación y estructura de la respuesta . . . . .	50
<b>5. Pruebas y resultados</b>	<b>54</b>
5.1. Motivación de las pruebas . . . . .	54
5.2. Corpus DIHANA . . . . .	55
5.3. Validación de los módulos del sistema . . . . .	57
5.3.1. Módulo de reconocimiento del lenguaje . . . . .	57
5.3.1.1. Descripción de la prueba . . . . .	57
5.3.1.2. Evaluación de la prueba . . . . .	58
5.3.2. Módulo de inferencia gramatical . . . . .	61
5.3.2.1. Descripción de la prueba . . . . .	61

<i>ÍNDICE GENERAL</i>	vi
5.3.2.2. Evaluación de la prueba . . . . .	64
5.3.3. Generación de la plantilla de respuesta . . . . .	65
5.3.4. Módulo de generación de respuesta . . . . .	67
5.3.4.1. Descripción de la prueba . . . . .	69
5.3.4.2. Evaluación de la prueba . . . . .	71
<b>6. Conclusiones</b>	<b>72</b>
6.1. Contribuciones . . . . .	73
6.2. Trabajo futuro . . . . .	76
<b>Bibliografía</b>	<b>78</b>



# Índice de figuras

2.1. Ejemplo de un diálogo con cinco intervenciones. . . . .	9
2.2. Arquitectura de un sistema de diálogo [ <a href="#">Llisterri et al. (2006)</a> ]. . . . .	10
4.1. Estructura general del Sistema de Diálogo. . . . .	36
4.2. Preparación y etiquetado de los datos de entrenamiento y prueba para un NER. . . . .	38
4.3. Ejemplo de un diálogo del corpus DIHANA donde se muestran las diferentes intervenciones del usuario y de la máquina. . . . .	39
4.4. Ejemplo de diálogo etiquetado del corpus DIHANA. . . . .	39
4.5. Etapa de entrenamiento de un NER. . . . .	40
4.6. Etapa de evaluación de un NER. . . . .	40
4.7. Secuencias de entidades semánticas obtenidas del reconocimiento del lenguaje del diálogo mostrado en la Figura 4.3. . . . .	42
4.8. Diagrama de estados del AFD para el lenguaje regular <i>Paridad</i> . . . . .	45
4.9. Esquema general del funcionamiento del módulo de generación de la plantilla de respuesta. . . . .	49
4.10. Esquema general del funcionamiento del Sistema de Diálogos en donde se muestra el módulo que construye y verifica la respuesta en lenguaje natural de manera escrita. . . . .	51
5.1. Ejemplo de un diálogo del corpus DIHANA; se muestran las diferentes intervenciones del usuario y de la máquina. . . . .	56

5.2. Ejemplo de una conversación etiquetada en inglés y la secuencia de entidades semánticas reconocidas por el módulo de reconocimiento de lenguaje. . . . .	58
5.3. Ejemplo de un Autómata Inferido con el modelo propuesto. . . . .	63
5.4. Autómata Inferido con el algoritmo de Inferencia, utilizando el 80 % de conversaciones del corpus DIHANA. . . . .	64
5.5. Autómata inferido con el modelo de inferencia gramatical entrenado con secuencias de entidades semánticas modeladas en el idioma Inglés .	65
5.6. Reconocimiento de la secuencia de entidades semánticas: “ <i>question schedules destination_city trip_class</i> ”, en el autómata inferido . . . . .	66
5.7. Respuesta inferida por el autómata: “ <i>ask_for_date</i> ”. . . . .	67
5.8. Reconocimiento de la etiqueta semántica “ <i>date</i> ” que corresponde al dialogo “ <i>El próximo sábado</i> ” . . . . .	68
5.9. Respuesta inferida por el autómata “ <i>check_schedules origin_city destination_date trip_class wait</i> ” que corresponde al diálogo: “ <i>le consulto horarios de trenes con salida en valencia destino pamplona saliendo el día 1 de mayo de 2004 en clase turista. Un momento por favor</i> ” . .	68
5.10. Respuesta inferida por el autómata en una segunda intervención del sistema “ <i>first departure_time arrival_time last departure_time2 arrival_time2 new query</i> ” que corresponde al diálogo: “ <i>El primero sale a las 7 y 3 de la mañana y llega a la una y 35 de la tarde y el lunes sale a las 10 y 3 de la noche y llega a las 7 de la mañana. Desea algo más</i> ”. . . . .	69
5.11. Se muestra la clase <i>RECORRIDO</i> con sus instancias correspondientes (recorridos). . . . .	70
5.12. Se muestra la ontología con las cinco clases. . . . .	70
5.13. Ejemplo de consulta a la ontología utilizando SPARQL. . . . .	71
5.14. Ejemplo de Respuesta de la consulta a la ontología, en la parte superior de cada columna de la respuesta se muestra el uso de las entidades semánticas. . . . .	71

# Índice de tablas

3.1. Clasificación de los diferentes modelos utilizados en la generación automática de diálogos. . . . .	34
4.1. Reconocimiento del lenguaje natural aplicado a la oración “ <i>Quisiera horarios de trenes para Madrid el próximo viernes</i> ” que forma parte del corpus DIHANA. . . . .	42
4.2. Ejemplo de la descripción de un lenguaje regular con alfabeto $\Sigma = \{a, b\}$ mediante su gramática regular derecha y su expresión regular equivalente. . . . .	44
4.3. Etiquetas y el valor asociado a cada etiqueta y su interpretación. . . .	52
5.1. Principales características del corpus DIHANA. . . . .	55
5.2. Exactitud obtenida en la identificación de tokens. . . . .	59
5.3. Resultados obtenidos de la evaluación de los dos NER. . . . .	60
5.4. Resumen de la validación cruzada a 10 pliegues para la prueba de identificación de secuencias de entidades semánticas en la gramática inferida. . . . .	65

# Capítulo 1

## Introducción

Hoy en día la computación está cada vez más presente en la vida cotidiana de las personas. Con respecto a este hecho, muchos investigadores comenzaron a pensar en formas de mejorar la *Interacción Humano-Computadora* (IHC) para que las aplicaciones y dispositivos de cómputo brinden a los usuarios una interacción más natural, teniendo en cuenta el contexto en el que se insertan los usuarios [[Harper et al. \(2008\)](#)].

La IHC es un campo de estudio dedicado a comprender, diseñar y evaluar sistemas de cómputo para su uso por o con humanos. La interacción, por definición, requiere comunicación entre humanos, computadoras y dispositivos. La IHC es actualmente una actividad de investigación y diseño muy extensa y diversa. La literatura se está expandiendo rápidamente, con cientos de publicaciones cada año y con la actividad de diferentes sociedades profesionales y reuniones ad hoc, principalmente en las disciplinas técnicas de ingeniería mecánica y eléctrica, ciencias de la computación y ciencias de control e inteligencia artificial.

El objetivo de las investigaciones en torno a la IHC es definir modelos que cubran las expectativas de los seres humanos con respecto a la interacción con las computadoras para guiar el diseño de éste y el desarrollo de algoritmos que permitan una interacción más natural y eficaz con los seres humanos. Con el avance de la Inteligencia Artificial (IA; del inglés *Artificial Intelligence: AI*), las investigaciones

se centran en una parte hacia la interacción física más segura y además sobre una interacción socialmente correcta, dependiente de criterios culturales.

Dentro de las aplicaciones en el área de la IHC se encuentran los sistemas de diálogo, los cuales tienen como objetivo facilitar la interacción natural entre una persona y un sistema informático. Estos tipos de sistemas se pueden definir como programas de computadora desarrollados para proporcionar servicios específicos a los seres humanos de la misma manera que si estos servicios fueron proporcionados por los mismos seres humanos, ofreciendo una interacción lo más natural y cómoda posible, en la cual el usuario interactúa con la computadora a través de diálogos.

En este trabajo de tesis, se propone un modelo de inferencia gramatical que sea capaz de inferir los diálogos de respuesta para una conversación dada entre un usuario y la computadora. Este modelo de inferencia gramatical es la parte medular del sistema de diálogos escritos de tipo pregunta-respuesta desarrollado en este trabajo de tesis. Cabe mencionar que para poder llevar a cabo el entrenamiento de dicho modelo se utiliza un corpus de dominio específico enfocado a la venta de boletos de trenes, el cual se describe con más detalle en la Sección 5.2.

## 1.1. Planteamiento de la investigación

La IHC constituye un campo de investigación amplio, es en este campo de investigación donde se enmarca el contenido de esta tesis doctoral. La principal motivación de este trabajo surge de la necesidad de lograr una IHC a través de diálogos escritos del tipo pregunta-respuesta de manera automática. Aunque en un sentido amplio, los diálogos pueden ser muy generales, en este caso en particular se trabajará sobre un dominio específico. Actualmente sigue siendo un desafío el crear sistemas autónomos que pueden interactuar sin problemas con los humanos y navegar de manera segura en un mundo cambiante e incierto. Existen muchos retos tecnológicos que resolver; uno de ellos es el de la comunicación entre el humano y la computadora a través de diálogos. El problema de investigación en este trabajo de tesis doctoral consiste en lograr la comunicación a través de diálogos en un dominio específico de

una manera natural y eficaz entre el humano y la computadora, utilizando técnicas de aprendizaje automático. Por ello, *se persigue el objetivo de crear un sistema de diálogo que posibilite la interacción entre los humanos y la computadora de la manera más natural posible*. Para lograrlo, ha sido necesario llevar a cabo tareas de investigación, desarrollo, e integración en diversos campos relacionados con la IHC: diálogos, procesamiento de lenguaje natural (PLN), etc.

## 1.2. Objetivos de la investigación

El objetivo general se centra en una propuesta de solución a la problemática planteada anteriormente y es el siguiente:

- *Desarrollar un modelo computacional para la generación automática de diálogos en un dominio específico, mediante el uso de técnicas de aprendizaje automático.*

### 1.2.1. Objetivos específicos:

1. Determinar el dominio de trabajo para la generación automática de diálogos.
2. Evaluar el estado del arte en el uso de clasificadores supervisados para la generación automática de diálogos
3. Etiquetar morfosintáctica y semánticamente el corpus de diálogos.
4. Implementar métodos para el preprocesamiento de datos que serán utilizados en el reconocimiento del lenguaje.
5. Implementar un método para inferir la gramática asociada a un conjunto de diálogos correspondientes a un dominio específico .
6. Diseñar pruebas para la evaluación del modelo computacional para la generación automática de diálogos en un dominio específico.

### 1.3. Preguntas de investigación

En base al problema planteado en este trabajo y a los objetivos propuestos se definen las siguientes preguntas de investigación:

- ¿En qué medida es posible generar diálogos automáticamente mediante el uso de técnicas basadas en aprendizaje automático?
- ¿El dominio de trabajo influye para la generación automática de diálogos?
- ¿Qué factores influyen en el método para la obtención de los datos de entrada (diálogos entre personas)?
- ¿De qué manera el etiquetado del corpus impacta en la generación de los diálogos?
- ¿En qué medida es posible generar pruebas para la evaluación del modelo computacional desarrollado?

### 1.4. Aportaciones de la investigación

La principal contribución de este trabajo consiste en desarrollar un modelo computacional que utiliza técnicas de aprendizaje automático para permitir la generación de diálogos coherentes y apropiados. La construcción de dicho modelo incluye los siguientes componentes:

- Un módulo de reconocimiento del lenguaje en el cual se diseña e implementa una técnica basada en el reconocimiento de entidades nombradas (NER por sus siglas en inglés) para transformar textos planos (correspondientes a las intervenciones de diálogos) a secuencias de entidades semánticas que puedan ser perfectamente entendibles por una gramática computacional.
- Un módulo de inferencia gramatical al reconocimiento de cadenas asociadas a las intervenciones del usuario y la generación de una plantilla de respuesta en un formato de secuencias de entidades semánticas.

- Un módulo para la generación de la respuesta al usuario que incluye la obtención de los valores de respuesta a partir de una ontología diseñada explícitamente para el caso de estudio.

## 1.5. Organización de la tesis

El presente trabajo de tesis se ha organizado de la siguiente manera. En el Capítulo 1 se presenta una introducción al tema de investigación, también se exponen el objetivo general y los objetivos específicos, las preguntas de investigación y las principales aportaciones de esta investigación. En el Capítulo 2 se presenta el marco teórico necesario para comprender el dominio del problema. En el Capítulo 3 se hace un recorrido sobre diferentes publicaciones, en donde se describe el estado del arte correspondiente a la evolución de los sistemas de diálogo, así como las técnicas utilizadas en la generación de diálogos. En el Capítulo 4 se describe el modelo del Sistema de Diálogo desarrollado, así como la metodología de experimentación. En el Capítulo 5 se muestran y discuten los resultados experimentales. Por último, en el Capítulo 6 se desarrollan las principales conclusiones derivadas de este trabajo de tesis doctoral, un resumen de las principales aportaciones, también se plantean las líneas de investigación y trabajos para el futuro.



# Capítulo 2

## Marco Teórico

En este capítulo se describen los conceptos teóricos necesarios para la comprensión del dominio del problema, iniciando con la interacción humano - computadora. Dado que en este trabajo de tesis se aborda el concepto de la interacción humano computadora a través de diálogos escritos del tipo pregunta-respuesta, es de particular relevancia describir terminología relacionada con los sistemas de diálogo y los elementos que lo componen. El enfoque propuesto en este trabajo de tesis doctoral para garantizar un reconocimiento adecuado de entidades semánticas hace necesario incluir dentro de este capítulo una descripción de los métodos de aprendizaje automático más utilizados en la literatura, en particular, se describen tres tipos de aprendizaje automático y algunas técnicas utilizadas en algoritmos de aprendizaje automático.

### 2.1. Interacción humano-computadora

En términos generales, es la disciplina que estudia el intercambio de información mediante software entre las personas y las computadoras. Esta disciplina se encarga del diseño, evaluación e implementación de los aparatos tecnológicos interactivos, estudiando el mayor número de casos que les pueda llegar a afectar [[Castro and Rodríguez \(2018\)](#)]. El objetivo es que el intercambio sea más eficiente: minimizar errores, incrementar la satisfacción, disminuir la frustración y, en definitiva, hacer

más productivas las tareas que rodean a las personas y las computadoras [Karray et al. (2008)].

Los principales componentes en la IHC que proponen [Muñoz et al. (2014)], son:

- *Usuario:* Hay que tener en cuenta que el ser humano tiene una capacidad limitada de procesar información; lo cual es muy importante considerar al hacer el diseño. El usuario se puede comunicar a través de cuatro canales de entrada/salida: visión, audición, tacto y movimiento. La información recibida se almacena en la memoria sensorial, la memoria a corto plazo y la memoria a largo plazo. Una vez recibida la información, ésta es procesada a través del razonamiento y de habilidades adquiridas.
- *Computadora:* El sistema utilizado puede afectar de diferentes formas al usuario dependiendo de los dispositivos de entrada/salida y de la velocidad de procesamiento.
- *Origen del proceso interactivo:* Es importante que haya una buena comunicación entre usuario y computadora, por este motivo la interfaz tiene que estar diseñada pensando en las necesidades del usuario.

Es importante destacar que en el contexto de la IHC el término “humano” no necesariamente hace referencia a un individuo puede referirse a un conjunto de éstos que tengan un perfil específico y que este trabajando de manera colectiva. Por otro lado el término “computadora” debe entenderse como un sistema genérico con poder de procesamiento computacional, cualesquiera que este fuera.

Por último el concepto de “interacción” es sumamente relevante ya que establece la manera en que interactúan o se relacionan los dos entes anteriormente mencionados. Existe una gran diversidad de mecanismos de interacción en función de los contextos en los cuales se realiza este proceso. Uno de los objetivos de IHC es precisamente el entender cómo se debería llevar a cabo este proceso. Así, un humano puede interactuar con una computadora a través de un teclado, una pantalla táctil, movimientos faciales, dispositivos de lectura encefalográfica, comandos de voz, etc.

Si bien, tal y como se mencionó anteriormente existe una diversidad de mecanismos de interacción entre los seres humanos y los equipos de computo, también hemos mencionado que nuestro interés de investigación se centra en el proceso del diálogo entre un usuario y una computadora, todo esto a través de intervenciones tipo pregunta-respuesta que se encuentran codificadas mediante secuencias de palabras escritas en lenguaje natural.

Por esta razón es conveniente que el lector comprenda el concepto de diálogo y sus componentes, los cuales se presentan en la siguiente sección.

## 2.2. Diálogo

La RAE define el diálogo como “La plática entre dos o más personas que alternativamente muestran sus ideas o afectos”. Otra definición que se puede encontrar es la que da la Wikipedia: “El diálogo es una forma oral y escrita en la que se comunican dos o más personajes en un intercambio de ideas por cualquier medio”. Finalmente, en WordReference, se encuentra la siguiente definición: “Conversación entre dos o más personas que intercambian el turno de palabra”.

Se puede decir entonces que el diálogo es un intercambio verbal entre dos o más personas o personajes, haciéndose alusión tanto a situaciones reales como imaginarias. El intercambio genera una variación constante en los roles de hablante y oyente. Cuando un interlocutor está dirigiendo la palabra cumple el rol de hablante, mientras el que escucha hace de oyente, cuando el otro interlocutor toma la palabra se da un intercambio de roles y así sucesivamente hasta el cierre de la conversación o el diálogo.

Este intercambio de roles genera “*intervenciones*” por parte de los participantes que conversan. En la Figura 2.1 se muestra un ejemplo de un dialogo corto con cinco intervenciones. Este diálogo está contextualizado en el marco de una interacción entre un usuario de un sistema de transporte y un equipo de computo que responde de manera automática a las preguntas y/o cuestionamientos de cada intervención de dicho usuario.

Este diálogo se presenta a modo de ejemplo y por lo tanto sólo muestra un conjunto pequeño de intervenciones.

<p><b>Intervención 1 (Usuario):</b> Hola buenas noches.</p> <p><b>Intervención 2 (computadora):</b> Buenas noches, ¿en que puedo servirle?</p> <p><b>Intervención 3 (Usuario):</b> ¿Cuánto cuesta el pasaje de Puebla a México en un autobús directo económico?</p> <p><b>Intervención 4 (computadora):</b> El precio de un viaje directo económico de Puebla a la ciudad de México es de 160 pesos. ¿Algo más en lo que lo pueda ayudar?</p> <p><b>Intervención 5 (Usuario):</b> No, gracias.</p> <p><b>Intervención 6 (computadora):</b> Estamos para servirle, hasta luego.</p>
--

Figura 2.1: Ejemplo de un diálogo con cinco intervenciones.

El diálogo anterior tiene un total de 6 intervenciones, tres para el usuario y tres para la computadora. La construcción de un sistema para reconocer lo que el usuario desea y transformarlo en una respuesta coherente requiere de una teoría conocida como “sitema de diálogo”, la cual se describe a continuación

## 2.3. Sistemas de diálogo

[Llisterri et al. (2006)] describe a los sistemas de diálogo o sistemas conversacionales ( del inglés *Spoken Language Systems: SLS*) como una tecnología concebida para facilitar la interacción natural mediante el habla entre una persona y una computadora. Constituyen una interfaz o un intermediario entre un usuario y un sistema de computo, que tiene la ventaja de no requerir el uso de una pantalla, un teclado o un ratón, de recurrir, en cambio, al medio de comunicación propio de los seres humanos. Además, [Llisterri et al. (2006)] dice que un sistema de diálogo consiste en una estructura modular en la que cada módulo se ocupa de unas determinadas tareas en interacción con todos los demás. La Figura 2.2 describe una arquitectura de un sistema de diálogo, el cual consta de un módulo de reconocimiento de voz que se ocupa de transformar la señal sonora en una representación que pueda procesar el módulo de análisis semántico; una vez realizado este análisis, interviene el módulo de gestión del diálogo, conectado a una interfaz que permite

la consulta a una base de datos y que envía la información recogida al módulo de generación de respuestas; este último, con la intervención de un conversor de texto en habla, facilitaría al usuario la información deseada. Como elemento central, estos autores incluyen el contexto del diálogo, que contiene información sobre el enunciado que se está procesando, la historia del diálogo y el estado actual de la interacción.

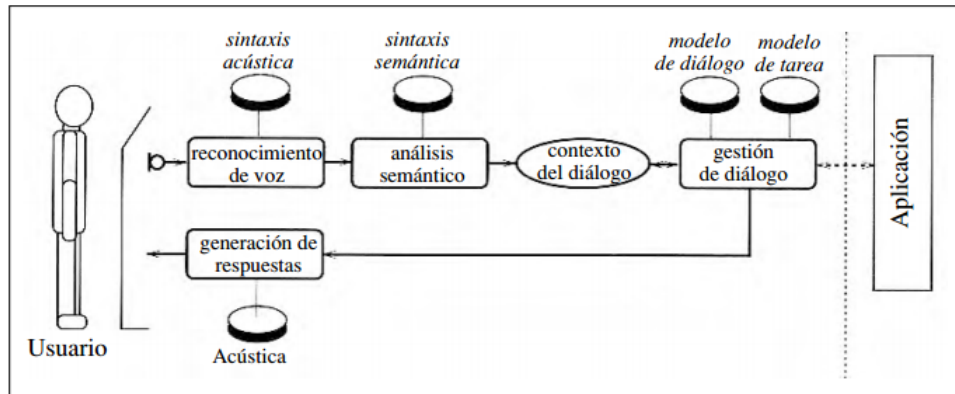


Figura 2.2: Arquitectura de un sistema de diálogo [Llisterri et al. (2006)].

Puede observarse cómo cada uno de los módulos de un sistema de diálogo está asociado a uno o más modelos que reflejan los conocimientos necesarios para garantizar la interacción del usuario y una aplicación de computo. Así, los módulos de reconocimiento y de análisis emplean la información procedente de los modelos de lenguaje, y el de gestión del diálogo recurre al modelo del diálogo y al modelo de la tarea.

Existen aproximaciones en la construcción de sistemas de diálogo que recurren a procedimientos basados en el conocimiento, estos utilizan reglas predeterminadas las cuales identifican patrones léxicos y gramaticales de las intervenciones que los usuarios hacen en lenguaje natural y a partir de ahí seleccionan una posible respuesta dentro de un banco de respuestas escritas de manera predeterminada. Ejemplos de este tipo de sistemas de diálogo son todos aquellos que están montados en una infraestructura basada en AIML, un lenguaje de marcas para inteligencia artificial. Su principal inconveniente es la incapacidad para aprender nuevos tipos de comportamientos en sus intervenciones durante un diálogo ya que tanto los patrones

asociados al reconocimiento de las preguntas, como las posibles respuestas son escritas y determinadas a priori, generando una convergencia inevitable a diálogos repetitivos y reduciendo la experiencia positiva del usuario.

Existen otros tipos de aproximaciones que son de mayor interés para este trabajo de investigación y que se encuentran basadas en técnicas de aprendizaje automático lo cual permite aprender a partir de conjuntos de diálogos previamente etiquetados. Esta aproximación es mucho más dinámica y da oportunidad a controlar el flujo de la conversación para generar diálogos mucho más coherentes.

Es importante enfatizar que si bien existen en la literatura diferentes arquitecturas para los sistemas de diálogo en este caso se han considerado tres grandes componentes: a) reconocimiento del lenguaje, b) inferencia gramatical, y c) generación de la respuesta.

Para el primer componente se requiere entender el concepto de entidades nombradas, en particular aquellas que tienen una carga de significado (entidades semánticas), y por esta razón en la Sección 2.4 se presenta a detalle el concepto de entidades nombradas a fin de que el lector tenga una claridad sobre el significado de las mismas. Tanto en el primer componente como en el segundo se utilizan textos almacenados en computadora que contienen información relevante para el proceso de aprendizaje. Este tipo de colecciones textuales se conoce como “corpus” y la descripción de este concepto se presenta en la Sección 2.5.

También se incluye una descripción moderada para el concepto de aprendizaje automático en la Sección 2.7.

Finalmente y como apoyo para entender los procedimientos para la generación de la respuesta se incluye en la Sección 5.11 una descripción al concepto de ontologías.

El objetivo ha sido evitar sobresaturar al lector con conceptos innecesarios y centrarlo en aquellos que sean exclusivaente necesarios para el entendimiento del resto del documento.

## 2.4. Entidades nombradas

En la extracción de información, una entidad nombrada es un objeto del mundo real, como personas, ubicaciones, organizaciones, productos, etc., que se puede denotar con un nombre propio. Puede ser abstracto o tener una existencia física. Entre los ejemplos de entidades nombradas se incluyen Benito Juárez, Estados Unidos Mexicanos, Volkswagen Golf o cualquier otra cosa que pueda nombrarse. Las entidades nombradas pueden simplemente ser vistas como instancias de entidades (por ejemplo, la ciudad de Oaxaca o Guadalajara son instancias de ciudad).

Desde una perspectiva histórica, el término Entidad Nombrada fue acuñado por [Grishman and Sundheim (1995)] durante la campaña de evaluación MUC-6 (Message Understanding Conferences) que contenía ENAMEX (expresiones de nombres de entidades, por ejemplo, personas, ubicaciones y organizaciones) y NUMEX (expresión numérica).

Una definición más formal se puede derivar del designador rígido de Saul Kripke<sup>1</sup>. En la expresión “entidad nombrada”, la palabra “nombrada” tiene como objetivo restringir el posible conjunto de entidades a aquellas para las que uno o varios designadores rígidos representan. Un designador es rígido cuando designa lo mismo en todos los mundos posibles. Por el contrario, los designadores flácidos pueden designar diferentes cosas en diferentes mundos posibles.

Como ejemplo, consideremos la oración: “Benito Juárez fue el presidente de los Estados Unidos Mexicanos”. Tanto “Benito Juárez” como “Estados Unidos Mexicanos” son entidades nombradas ya que se refieren a objetos específicos. Sin embargo, “presidente” no es una entidad con nombre ya que puede usarse para referirse a muchos objetos diferentes en mundos diferentes (en diferentes períodos presidenciales que se refieren a diferentes personas, o incluso en diferentes países u organizaciones que se refieren a diferentes personas). Los designadores rígidos suelen incluir nombres propios y ciertos términos naturales, como especies biológicas y sustancias.

También existe un acuerdo general en la comunidad de Reconocimiento de

---

<sup>1</sup>Kripke, Saul: Naming and Necessity. Harvard University Press, 1980.

Entidades Nombradas para considerar como entidades nombradas expresiones temporales y numéricas tales como cantidades de dinero y otros tipos de unidades.

## 2.5. Corpus

Definir el concepto de corpus, tal y como se usa hoy en día en el ámbito de la lingüística o lexicografía de corpus, o en la lingüística computacional en general, no es tan sencillo como podría parecer a primera vista. En principio, se puede llamar corpus a cualquier colección que contenga más de un texto (corpus como cuerpo textual).

Estas implicaciones se hacen patentes en las diferentes definiciones de corpus propuestas en los últimos años como se muestra en [Sadick Ngula (2014)] que referencia a Leech (1992), a Francis (1982), a Sinclair (1991) y a Atkins, Clear y Ostler (1992) [M. Chantal (2002)], los cuales introducen el concepto de corpus de la siguiente forma:

- *Leech (1992)* On the face of it, a computer corpus is an unexciting phenomenon: a helluva lot of text, stored on a computer (“*A primera vista, un corpus de computadora es un fenómeno poco emocionante: una gran cantidad de texto, almacenado en una computadora*”), donde se refleja que, aunque sea de un modo bastante simplista, podemos considerar que un corpus no es más que una colección de texto en formato magnético, aunque Leech completa su definición recalando que la habilidad que poseen los ordenadores para buscar, recuperar, ordenar y hacer cálculos sobre cantidades masivas de texto nos ha brindado la oportunidad de comprender y de explicar el contenido de esos corpus de formas que no eran imaginables en la era que él denomina “pre-computacional”. De hecho, dado que los avances tecnológicos van tan unidos al desarrollo de la lingüística de corpus tal y como hoy en día la conocemos, Leech argumenta que debe denominarse Computer Corpus Linguistics, ya que el término “lingüística de corpus” se usaba antes del advenimiento de las computadoras digitales.



- *Francis (1982)* lo define como sigue: a corpus is a collection of texts assumed to be representative of a given language, dialect, or other subset of a language to be used for linguistic analysis (“*un corpus es una colección de textos que se supone son representativos de un idioma, dialecto u otro subconjunto de un idioma para ser utilizado para el análisis lingüístico*”).
- *Sinclair (1991)* lo define como: a corpus is a collection of naturally-occurring language text, chosen to characterize a state or variety of a language (“*un corpus es una colección de texto de lenguaje natural, elegido para caracterizar un estado o variedad de un idioma*”).
- La definición que ofrecen *Atkins, Clear y Ostler (1992)* añade otro aspecto esencial en la creación de un corpus: éste debe ser construido de acuerdo con una serie de criterios explícitos: a corpus is a subset of an ETL (Electronic Text Library) built according to explicit design criteria for a specific purpose (“*un corpus es un subconjunto de una biblioteca de texto electrónica construida de acuerdo con criterios de diseño explícitos para un propósito específico*”).
- Quizá la definición más estandarizada la ofrece el grupo de trabajo dedicado a los corpus textuales de *EAGLES (Expert Advisory Group on Language Engineering Standards) (1996a:4)*: Corpus: A collection of pieces of language that are selected and ordered according to explicit linguistic criteria in order to be used as a sample of the language (“*Una colección de piezas de lenguaje que se seleccionan y ordenan de acuerdo con criterios lingüísticos explícitos para ser utilizados como muestra del idioma*”). En esta definición se recogen tres aspectos fundamentales que se deben tener en cuenta en la definición de los corpus. Un corpus debe estar compuesto por textos producidos en situaciones reales y la inclusión de los textos que componen el corpus debe estar guiada por una serie de criterios lingüísticos explícitos para asegurar que pueda usarse como muestra representativa de una lengua.

Como parte de procesos de la creación de un corpus es necesario considerar

una técnica sumamente utilizada conocida como Mago de Oz, la cual se describe a continuación.

## 2.6. Técnica del Mago de Oz

La técnica del Mago de Oz (*“Wizard of Oz”*) es una forma eficiente de examinar la interacción del usuario con las computadoras y facilitar el rápido desarrollo iterativo de la redacción y la lógica del diálogo. La técnica requiere dos máquinas unidas, una para el usuario y otra para el experimentador. El experimentador (el “asistente”), simulando ser una computadora, responde a las consultas de los usuarios directamente o presionando las teclas de función a las que se han asignado mensajes comunes.

Esta técnica del Mago de Oz se basa en la historia de *“El maravilloso mago de Oz”*<sup>2</sup>. En la historia, el mago produjo imágenes asombrosas de sí mismo al manipular un conjunto de controles mientras estaba oculto detrás de una cortina.

La técnica del Mago de Oz es un proceso que permite al usuario interactuar con una interfaz sin saber que las respuestas están siendo generadas por un humano en lugar de una computadora. Esta técnica permite a los investigadores probar un concepto haciendo que un profesional, el ‘Moderador’, dirija la sesión cara a cara con cada usuario, mientras que otro profesional, el ‘Mago’, controla las respuestas enviadas al usuario a través del dispositivo.

Una de las principales aplicaciones de esta técnica se relaciona con los sistemas de diálogo y especialmente en el tema de errores en la comunicación, en el habla espontánea, o en la definición de escenarios entre otros. Otra aplicación importante es en la recolección de corpus con fines generales o específicos, como es el caso del corpus DIHANA que se emplea en este trabajo, el cual fue construido precisamente mediante la técnica del Mago de Oz [Miguel et al. (2003)].

---

<sup>2</sup>Libro de literatura infantil escrito por Lyman Frank Baum 1900.

## 2.7. Aprendizaje automático

El Aprendizaje Automático (AA; del inglés *Machine Learning: ML*) es el subcampo de las ciencias de la computación y una rama de la IA que tiene como objetivo desarrollar técnicas que permitan a las computadoras aprender. De forma más concreta, se trata de crear algoritmos capaces de generalizar comportamientos y reconocer patrones a partir de una información suministrada en forma de ejemplos. Es, por lo tanto, un proceso de inducción del conocimiento, es decir, un método que permite obtener un enunciado general a partir de enunciados que describen casos particulares<sup>3</sup>. Una de las tareas del AA es intentar extraer conocimiento sobre algunas propiedades no observadas de un objeto, basándose en las propiedades que sí han sido observadas de ese mismo objeto (o incluso de propiedades observadas en otros objetos similares). En palabras más llanas, predecir comportamiento futuro a partir de lo que ha ocurrido en el pasado<sup>4</sup>.

[[Mitchell \(1997\)](#)] proporcionó una definición ampliamente citada y más formal de los algoritmos estudiados en el campo del aprendizaje automático: “*Se dice que un programa de computadora aprende de la experiencia  $E$  con respecto a una clase de tareas  $T$  y la medida de rendimiento  $P$  si funciona en tareas en  $T$ , medido por  $P$ , mejora con la experiencia  $E$ ”.*

Ejemplo: Jugando al ajedrez.

$E$  = La experiencia de jugar muchos juegos de ajedrez.

$T$  = La tarea de jugar ajedrez.

$P$  = La probabilidad que el programa gane el próximo partido.

[[Alpaydin \(2010\)](#)] menciona que el modelo puede ser predictivo para hacer predicciones en el futuro, o descriptivo para obtener conocimiento de los datos, o ambos.

---

<sup>3</sup>Andrew Ng: Profesor adjunto en la Universidad de Stanford, director de Google Brain, cofundador de Coursera y vicepresidente de Baidu Research.

<sup>4</sup>Yoshua Bengio: Profesor en el Departamento de Ciencias de la Computación e Investigación de Operaciones de la Universidad de Montreal y director científico del Instituto de Montreal para Algoritmos de Aprendizaje (MILA).

### 2.7.1. Métodos de aprendizaje automático

Dentro del AA existen tres métodos de aplicación diferenciados, los cuales se explican a continuación.

- *Método de Regresión:* Este método se utiliza para predecir el valor de un atributo continuo. Consiste en encontrar la mejor ecuación que atraviese de forma óptima un conjunto de puntos ( $n$ -dimensiones). Se utiliza cuando la precisión no es crítica y el número de variables es pequeño.
- *Método de Clasificación:* Método utilizado para predecir un resultado de un atributo con valor discreto ( $a, b, c, \dots$ ) dadas las características ( $X_0, X_1, X_2, X_3, \dots, X_n$ ). El método simple de clasificación es el binario, donde se clasifica un registro de variables de entrada en 1 o 0. La clasificación múltiple es una extensión de la clasificación binaria.
- *Método de Agrupamiento:* Este método se utiliza cuando se necesita agrupar las instancias de datos, pero no se conocen previamente las categorías. Esta agrupación permite construir grupos (cluster) coherentes de instancias teniendo en cuenta las variables de los datos.

### 2.7.2. Tipos de aprendizaje automático

[Alpaydin (2010)] define tres tipos de algoritmos de AA:

- *Aprendizaje supervisado:* el objetivo es aprender un mapeo desde la entrada hasta una salida cuyos valores correctos son proporcionados por un supervisor, donde la base de conocimientos del sistema está formada por ejemplos etiquetados a priori (es decir, ejemplos de los que sabemos su clasificación correcta). Consta de 2 fases, una de entrenamiento y otra de pruebas.
- *Aprendizaje no supervisado:* en el aprendizaje no supervisado, no existe tal supervisor y sólo tenemos datos de entrada. El objetivo es encontrar las regularidades en la entrada. Hay una estructura en el espacio de entrada de

forma tal que ciertos patrones ocurren más a menudo que otros, y queremos ver qué sucede generalmente y qué no. En estadística, esto se llama estimación de densidad.

- *Aprendizaje por refuerzo*: en algunas aplicaciones, la salida del sistema es una secuencia de acciones. En tal caso, una sola acción no es importante; lo importante es la política que es la secuencia de acciones correctas para alcanzar la meta. No existe la mejor acción en ningún estado intermedio; una acción es buena si es parte de una buena política. En tal caso, el programa de AA debería ser capaz de evaluar la bondad de las políticas y aprender de las secuencias de acciones buenas del pasado para poder generar una política.

## 2.8. Campo Aleatorio Condicional

Un campo aleatorio condicional (CAC; del inglés *Conditional Random Field: CRF*) lo definen [Lafferty et al. \(2001\)](#) y [Pinto et al. \(2003\)](#), como un modelo estocástico utilizado habitualmente para etiquetar y segmentar secuencias de datos o extraer información de documentos. En algunos contextos también se le denomina Campo Aleatorio de Markov (CAM; del inglés *Markov Random Field: MRF*).

Dada una secuencia de datos (palabras) observados de longitud  $n$  ( $O_1, \dots, O_n$ ) este modelo asigna una etiqueta  $S_i$  para cada elemento  $O_i$ , donde  $S$  es un conjunto de estados en una máquina de estados finitos, cada uno de los cuales corresponde a una etiqueta  $l \in L$ . y  $(S_1, S_2, \dots, S_n)$  es la secuencia de estados en  $S$  que corresponden a las etiquetas asignadas a las palabras en la secuencia de entrada  $O$ . Aunque presenta similitudes con los modelos ocultos de Markov (MOM; del inglés *Hidden Markov Models: HMM*), estos son modelos generativos que conforman conjuntamente la distribución de probabilidad de las etiquetas ó estados ( $S$ ) y las observaciones ( $O$ ),  $P(S, O)$  mientras que los campos aleatorios condicionales modelan la probabilidad de la secuencia correcta de etiquetas condicionadas por las observaciones,  $P(S, O)$  es decir, son modelos discriminativos.

Un campo aleatorio condicional se puede representar con un grafo no dirigido  $G = (V, E)$  en el que cada vértice  $V$  represente una variable aleatoria cuya distribución de probabilidad debe ser deducida, y cada arista  $E$  indique una dependencia entre las variables de los vértices que conecta. El grafo satisface la propiedad de Markov extendida a grafos:

$$P(S_i | O, S_j; i \neq j) = P(S_i | O, S_j; S_i \sim S_j)$$

donde  $\sim$  significa que los vértices  $S_i$  y  $S_j$  están conectados por una arista, es decir son vecinos en  $G$ . En cuanto a los datos  $O_i$ , también llamados observaciones, lo más frecuente es que sean también una secuencia. Además, es frecuente que cada  $O_i$  sea un vector, no un valor escalar, en cuyo caso tendríamos observaciones multidimensionales.

El grafo puede tener una estructura arbitrariamente compleja, aunque lo más común es que sea una cadena o un rejilla". En una cadena, cada vértice está únicamente conectado con el vértice predecesor y con sus sucesores (se asume que los vértices están ordenados). En una rejilla, cada vértice está conectado con otros 4, excepto en los extremos; un vértice  $S_{ij}$  estará conectado con  $S_{i,j-1}$ ,  $S_{i,j+1}$ ,  $S_{i-1,j}$  y  $S_{i+1,j}$ . En el caso de la cadena la propiedad de Markov puede reescribirse de la siguiente forma:

$$P(S_i | O, S_j; i \neq j) = P(S_i | O, S_j; S_{i-1}, S_{i+1})$$

Estos modelos necesitan ser entrenados con  $N$  muestras  $(O^{(i)}, S^{(i)})_1^N$  donde cada  $(O^{(i)} = (o_1^{(i)}, o_2^{(i)}, \dots, o_t^{(i)}))$  es una secuencia de entradas o conjunto de observaciones, así como las etiquetas asociadas a esas observaciones  $(S^{(i)} = (s_1^{(i)}, s_2^{(i)}, \dots, s_t^{(i)}))$ . El modelo extrae un conjunto de características  $f(i, S_i, S_{i+1})$  y  $g(i, S_i, O)$  que representan las dependencias existentes entre diferentes estados y entre estos y la secuencia de observaciones. Al contrario que en los modelos ocultos de Markov en donde cada estado  $S_i$  depende únicamente de la observación  $O_i$ , aquí cada estado puede depender de varias observaciones al mismo tiempo, incluso de la secuencia completa si

fuese necesario. En el entrenamiento del modelo, éste asigna unos pesos a cada una de esas características, indicando su relativa importancia según el caso.

## 2.9. Inferencia Gramatical

La Inferencia Gramatical (IG) es un campo de investigación bien establecido en inteligencia artificial que data de los años 60. [Gold (1967)] originó este estudio e introdujo la noción de *identificación del lenguaje en el límite*. Desde su trabajo seminal, ha habido una cantidad notable de trabajo para establecer una teoría de la inferencia gramatical, para encontrar métodos efectivos y eficientes para inferir gramáticas, y para aplicar esos métodos a problemas prácticos. El comienzo de las investigaciones acerca del aprendizaje de gramáticas puede situarse a mediados de los años 50 con el desarrollo de las gramáticas formales para modelizar la estructura del lenguaje natural [Chomsky (1957)], [Chomsky (1969)]. La inferencia gramatical se ha investigado, de manera más o menos independiente, en muchos campos de investigación, incluido el aprendizaje automático, la teoría del aprendizaje computacional, el reconocimiento de patrones, la lingüística computacional, las redes neuronales, la teoría del lenguaje formal, la recuperación de información, el diseño de lenguajes de programación, la traducción y compilación, los lenguajes gráficos, la comunicación hombre-máquina y muchos otros, todos ellos de innegable interés. La inferencia gramatical es parte del campo del aprendizaje automático, concretamente pertenece a la *inferencia inductiva*, es decir sistemas que tratan de extraer reglas generales a partir de ejemplos. En el caso concreto de la inferencia gramatical, son reglas de reescritura en el sentido de la Teoría de Lenguajes Formales, o una extensión natural de ellas [Becerra-Bonache (2008)].

La inferencia gramatical se define como el problema de inducir las reglas gramaticales o producciones que caracterizan una gramática a partir de un conjunto de ejemplos de entrenamiento [Fu (1982)].

El proceso de inferencia gramatical lo podemos sintetizar de la siguiente manera:

1. Disponer de pares de ejemplos  $(a_i, b_i)$ , donde  $i = 1 \dots, n$ , siendo  $n$  el número total de ejemplos disponibles,  $a_i$  una secuencia de símbolos terminales  $a_i \in T^*$ , donde  $T$  es el conjunto de símbolos terminales pertenecientes a un alfabeto  $\Sigma$  y  $b_i$  un entero,  $b_i \in \{0, 1\}$ , tal que si la secuencia  $a_i$  pertenece al lenguaje entonces  $b_i = 1$  (ejemplo positivo) y si no pertenece  $b_i = 0$  (ejemplo negativo).
2. Obtener, mediante algún método, las producciones asociadas a la gramática que genera el lenguaje al que pertenecen los ejemplos positivos del conjunto de aprendizaje. El proceso de extracción de la gramática  $G$  lleva consigo los siguientes pasos:
  - Aprender el conjunto de ejemplos
  - Extraer mediante algún método el autómata de estados finitos que reconoce el lenguaje asociado a los ejemplos de aprendizaje.
  - Extraer las producciones de la gramática asociadas al autómata anterior.

Dentro de los métodos de inferencia inductiva existen dos tipos básicos, los:

- *Constructivos*, que van construyendo una nueva hipótesis con cada nuevo ejemplo.
- *Enumerativos*, que asumen que es posible enumerar las hipótesis. A cada nuevo ejemplo, un método enumerativo buscará secuencial y exhaustivamente en la lista de hipótesis la primera que sea compatible con todos los ejemplos presentados.

Así también en inferencia inductiva se utilizan dos métodos de presentación de ejemplos [[Gold \(1967\)](#)]:

- *Presentación positiva* del lenguaje  $L$ : es una sucesión de elementos de  $L$  (muestras positivas).
- *Presentación completa* del lenguaje  $L$ : es una sucesión de elementos de  $L$  y de su complementario (muestras positivas y muestras negativas), marcados



para indicar su pertenencia o no a  $L$ . Todas las cadenas de  $V^*$  aparecen en la secuencia.

Ambos métodos son utilizados en inferencia gramatical, aunque básicamente se emplee la *presentación positiva* debido a que los métodos enumerativos no son lo suficientemente rápidos, y los métodos constructivos existentes no permiten la utilización de muestras negativas.

Como ya se mencionó anteriormente, la IG consiste en la inducción de las producciones asociadas a la gramática que responde al conjunto de ejemplos de aprendizaje. El objetivo es aprender el lenguaje correcto a partir de un conjunto de instancias de entrenamiento. En relación con este objetivo el resultado más destacable es que resulta imposible aprender el lenguaje correcto (o gramática) únicamente a partir de ejemplos positivos. [Gold (1967)] demuestra que, aunque se le proporcionen a un programa infinitas secuencias de ejemplos positivos, el programa no puede determinar en un tiempo finito una gramática que genere el lenguaje libre de contexto correcto. Esto es debido a que existen muchos lenguajes posibles que podemos colocar entre dos extremos. Un extremo lo constituye el lenguaje más general (lenguaje universal), que contiene todas las posibles palabras. El otro extremo es el lenguaje más específico, que contiene exclusivamente las secuencias que han sido proporcionadas como ejemplos positivos.

## 2.10. Ontología

El término *Onto-Logía* viene del término griego *onto* verbo “ser, estar” y de *logía* “ciencia, estudio, teoría”. Entonces se puede definir una ontología como una especificación formal de conceptos y relaciones entre los mismos de forma jerarquizada basada en el conocimiento, restringidos por axiomas. El término ontología tiene su origen en la filosofía. Para su creador, Cristián Wolf <sup>5</sup>, es la parte de la filosofía que se ocupa del examen de las propiedades de los seres: existencia, concebía la existencia desde una explicación sistemática, posibilidad, etcétera.

---

<sup>5</sup>Christian Wolff matemático, físico y filósofo alemán (1679 a 1754).

A principios de los 90, [Gruber (1993)], en su trabajo “*Toward Principles for the Design of Ontologies Used for Knowledge Sharing*” introduce una definición deliberada de ontología como un término técnico en ciencias de la computación. Gruber define una ontología como una especificación de una conceptualización y la define como sigue: *Una ontología es una descripción (como una especificación formal de un programa) de los conceptos y relaciones que pueden formalmente existir para un agente o comunidad de agentes.* Esta definición es consistente con el uso de ontología como un conjunto de definiciones conceptuales, pero más generales. Y esto es un sentido diferente de la palabra ontología utilizada en filosofía.

Las ontologías contemporáneas comparten muchas similitudes estructurales, indiferente al lenguaje en el cual ellos fueron expresados. La mayoría de las ontologías describen individuos (instancias), clases (conceptos), atributos y relaciones. Los componentes más comunes de una ontología son:

- *Individuos*: instancias u objetos (lo básico u objetos de “bajo nivel”).
- *Clases*: conjuntos, colecciones, conceptos, clases en programación, tipos de objetos, o tipos de cosas.
- *Atributos*: aspectos, propiedades, rasgos, características, o parámetros que objetos (y clases) pueden tener.
- *Relaciones*: formas en la cual clases y los individuos se pueden relacionar unos con otros.
- *Funciones*: Complejas estructuras formadas de cierta relación que pueden ser usadas en lugar de un término individual en una declaración.
- *Restricciones*: establecen descripciones formales de lo que debe ser verdad con el objetivo de que alguna aserción pueda ser aceptada como entrada.
- *Reglas*: Declaraciones con forma de oraciones **si-entonces** (antecedente-consecuente) que describen inferencias lógicas que puede ser derivables de una aserción en una forma particular.

- *Axiomas*: aserciones (incluyendo reglas) en una forma lógica que juntos incluyen toda la teoría que la ontología describe en su dominio de aplicación. Esta definición es diferente de los “axiomas” en gramáticas generadas y forma lógica. En esas disciplinas, los axiomas solamente incluyen declaraciones especificadas como un conocimiento a priori. En las ontologías, los “axiomas” también incluyen teorías derivadas de declaraciones axiomáticas.
- *Eventos*: los cambios de los atributos o relaciones.

Entre los lenguajes formales comunmente usados para representar las ontologías, se destacan los siguientes:

- *RDF* (Resource Description Framework). Marco de descripción de recursos desarrollado por W3C y basado en XML. En este lenguaje las relaciones entre dos objetos se establecen mediante el nombre de la relación y dichos elementos, similar a las redes semánticas.
- *OWL*. Lenguaje basado en XML y RDF, pertenece a la WC3 y es el estándar actual. Puede representar los elementos de lógica descriptiva. Además, tiene mayor capacidad expresiva. Es el más usado en Internet, estando sus elementos definidos con las fuentes de RDF. Tiene tres variantes según la complejidad que se necesite especificar, se muestran en orden ascendente de complejidad: OWL Lite, OWL DL y OWL Full.
- *CycL*. Desarrollado por Doug Lenat Cyc<sup>6</sup> en un proyecto basado en la inteligencia artificial, por ello es más bien un lenguaje declarativo basado en lógica de predicados de primer orden, añadiendo las extensiones para operadores. Es un sistema de código abierto, y sus principales características son la utilización de constantes para representar conceptos y representación de jerarquía, así como las normas en las que se apoyan la relaciones entre conceptos.
- *DOGMA*. Desarrollado en Vrije Universiteit Brussel Starlab, cuyo principal objetivo es solventar el problema lingüístico de las ontologías, adaptándolas

---

<sup>6</sup>Doug Lenat fundador, presidente y director ejecutivo de Cycorp en Austin, Texas.

a una independencia del idioma. Para ello realiza una separación del dominio con respecto a la conceptualización de su aplicación. Se puede considerar a este lenguaje como una mezcla entre RDF y OWL, teniendo en cuenta la diferenciación que hace con el nivel conceptual y el nivel del idioma.

- *OIL*. Fue desarrollado como proyecto de investigación por Dieter Fensel<sup>7</sup>, Frank van Harmelen<sup>8</sup> y Ian Horrocks<sup>9</sup>. Está basado en descripciones lógicas y en Marcos. Además, dentro de los lenguajes de consulta de ontologías destacan: SPARQL (SPARQL Protocol and RDF Query Language). Este lenguaje derivado de RDF permite consultas basadas en tres modelos, conjunciones, disyunciones, y de forma opcional, patrones. Permite la definición de prefijos y se pueden realizar consultas muy específicas. Su principal función es su utilización como lenguaje de consulta en la web semántica.

Dentro de las herramientas que existen para gestionar una ontología se puede destacar *Protégé*, Esta herramienta, fue creada en la universidad de Stanford y se utiliza para el desarrollo de Ontologías y Sistemas basados en el conocimiento por medio de una interfaz de usuario que facilita la creación de estructuras de frames con clases, slots e instancias de una forma integrada.

---

<sup>7</sup>Dieter A. Fensel: director del instituto de investigación DERI Innsbruck en la Universidad de Innsbruck, Austria.

<sup>8</sup>Frank van Harmelen: profesor de Representación y Razonamiento del Conocimiento en el departamento de Ciencias de la Computación (Facultad de Ciencias) de la Vrije Universiteit Amsterdam.

<sup>9</sup>Ian Horrocks: Editor en jefe del Journal of Web Semantics, presidente de la Semantics Web Science Association (SWSA), Fundador de Oxford Semantic Technologies.

# Capítulo 3

## Estado del Arte

En este capítulo se describen trabajos relacionados con la generación automática de diálogos. El objetivo principal de este capítulo es el de destacar los métodos, técnicas y herramientas para el dominio de los datos empleados para la generación de diálogos. La información encontrada en la literatura se presenta clasificada de acuerdo con los modelos empleados en cada uno de los trabajos analizados en el desarrollo de esta tesis de investigación. En particular, se consideraron los siguientes tres modelos:

- Modelos basados en reglas probabilísticas
- Modelos basados en procesos estocásticos
- Modelos basados en redes neuronales

A continuación se presentan de manera ordenada los trabajos de investigación considerados como los más relevantes encontrados en la literatura para cada una de las clasificaciones anteriormente mencionadas.

### 3.1. Modelos basados en reglas probabilísticas

Dentro de los trabajos que hacen uso del modelo de reglas probabilísticas que tienen que ver con la generación de diálogos podemos citar el trabajo de [[Konstantopoulos \(2010\)](#)] donde se propone una arquitectura novedosa para los sistemas de

diálogo afectivo y multimodal que permite un control explícito sobre los rasgos de personalidad que se quiere que el sistema exhiba. Más específicamente, acercarse a la personalidad como un medio de sintetizar diferentes y posiblemente conflictivos modelos de adaptación en un modelo general que se utilizará para impulsar los componentes de interacción del sistema. Además, esta síntesis se realiza en presencia de conocimiento de dominio, de manera que la estructura del dominio y las relaciones influyen en el cálculo de los resultados.

En el trabajo de [Lison (2012)] se muestra cómo representar la estructura subyacente de modelos probabilísticos para el diálogo utilizando reglas probabilísticas. Estas reglas se definen como asignaciones estructuradas sobre variables del estado de diálogo, especificadas usando condiciones y efectos de alto nivel. Estas reglas pueden incluir parámetros tales como probabilidades de efecto o utilidades de acción. Las reglas probabilísticas permiten al diseñador del sistema explotar poderosas generalizaciones en la especificación del dominio de diálogo sin sacrificar la naturaleza probabilística del modelo. El marco es muy general y puede expresar un amplio espectro de modelos, desde modelos clásicos totalmente estimados a partir de datos a aquellos que incorporan un rico conocimiento previo. La elección del modelo dentro de este espectro es, por lo tanto, esencialmente una decisión de diseño dependiente de las disponibilidades relativas de datos de entrenamiento y conocimiento de dominio. También se presentan algoritmos para construir redes bayesianas correspondientes a la aplicación de las reglas y para estimar sus parámetros a partir de datos utilizando la inferencia bayesiana. El enfoque presentado se ha implementado en un sistema de diálogo oral para la interacción humano-robot y se ha validado en una tarea de aprendizaje de políticas basada en un conjunto de datos de Wizard-of-Oz. Los resultados empíricos han demostrado que la estructura de reglas permite que el algoritmo de aprendizaje converja más rápido y con mejor rendimiento de generalización.

De igual forma [Litman and Forbes-Riley (2014)] presentan una evaluación de un sistema de diálogo oral que detecta y se adapta a la desconexión del usuario y a la incertidumbre en tiempo real. Comparan esta versión de su sistema con una

versión que se adapta sólo a la desconexión del usuario y a una versión que ignora por completo la desconexión del usuario y la incertidumbre. Encuentran un aumento significativo en el éxito de la tarea al comparar ambas versiones de adaptación de su sistema a su línea base no adaptativa, pero sólo para los usuarios masculinos. Su evaluación examina el impacto de la adaptación a diferentes números de estados afectivos en el éxito de la tarea, y también examina las interacciones con el género del usuario. Sin embargo, esta evaluación sólo se llevó a cabo en un escenario *Wizard of Oz*, donde un humano oculto reemplazó el reconocimiento de voz, el análisis semántico y los componentes de detección de afecto de su sistema de diálogo.

### 3.2. Modelos basados en procesos estocásticos

En esta sección se describen algunos trabajos que utilizan un modelo basado en procesos estocásticos para la generación de diálogos, conocido como proceso de decisión de Markov (del inglés *Markov Decision Process: MDP*) y otros más que utilizan un modelo de proceso de decisión de Markov parcialmente observable (del inglés *Markov Decision Process Partially Observable: POMDP*).

En el caso de [Barlier et al. (2015)] proponen un marco original para modelar los diálogos hablados de hombre-máquina para tratar la coadaptación entre usuarios y sistemas de diálogo hablados en tareas no cooperativas. La conversación es modelada como un juego estocástico: tanto el usuario como el sistema tienen sus propias preferencias, pero tienen que llegar a un acuerdo para resolver una tarea no cooperativa. Ellos son entrenados conjuntamente para que el gestor del diálogo aprenda la estrategia óptima contra el mejor usuario posible. Los resultados obtenidos por simulación muestran que se aprenden estrategias no triviales y que este marco es adecuado para el modelado del diálogo.

En los trabajos de [Misu et al. (2012)], [Pietquin (2013)], [Lison (2013)], [Png and Pineau (2011)], [Mahadik (2017)] y [Young et al. (2013)] utilizan el aprendizaje por refuerzo (del inglés *Reinforcement Learning: RL*) para aprender las políticas de diálogo de preguntas y respuestas para una aplicación del mundo real. En el

trabajo de [Misu et al. (2012)] se analiza un corpus de interacciones de visitantes del museo con dos personajes virtuales que sirven como guías en el Museo de Ciencias de Boston, con el fin de construir un modelo realista de comportamiento del usuario al interactuar con estos personajes. Se construye un usuario simulado sobre la base de este modelo y se utiliza para aprender la política de diálogo de los personajes virtuales usando RL. En el trabajo de [Lison (2013)] las contribuciones son dobles. En primer lugar, demostrar cómo aplicar el aprendizaje por refuerzo Bayesiano (del inglés *Bayesian Reinforcement Learning: BRL*) basado en modelos para aprender el modelo de transición de un dominio de diálogo. También se comparan dos enfoques de modelado en el contexto de un escenario humano-robot en el que se instruye a un robot Nao a moverse y recoger objetos. Los resultados empíricos muestran que el uso de representaciones estructuradas permite al algoritmo de aprendizaje converger más rápido y con mejor desempeño de generalización. En todos estos trabajos la idea clave es modelar la gestión del diálogo como un proceso de decisión de Markov o un proceso de decisión de Markov parcialmente observable, y dejar que el sistema aprenda por sí mismo la mejor acción a realizar en cada posible situación conversacional mediante interacciones repetidas con un usuario (real o simulado).

Ahora bien, en la investigación de [Iijima and Kobayashi (2016)] plantean que, en los últimos años, con la difusión de los robots domésticos, la necesidad de mejorar las capacidades de comunicación de esos robots a las personas ha ido en aumento. El objetivo de este estudio es construir un marco para un sistema de diálogo que se ocupe de la información multimodal que un robot observa. Ellos aplican POMDP para modelar la interacción multimodal entre un ser humano y un robot. A través de sus experimentos, han confirmado que el marco propuesto funciona correctamente y logra una interacción multimodal eficaz con un robot.

### 3.3. Modelos basados en redes neuronales

En esta sección se describen algunos trabajos que utilizan técnicas de aprendizaje automático para la generación de diálogos, la mayoría de estos trabajos utilizan



diferentes modelos de redes neuronales, las cuales han ido evolucionando muy rápido.

En el trabajo de [Dusek and Jurčicek (2016)] presentan un nuevo sistema de generación de lenguaje natural para sistemas de diálogo oral capaces de incorporar (adaptar) la forma de hablar de los usuarios, proporcionando respuestas contextualmente apropiadas. El generador se basa en una red neuronal recurrente (del inglés *Recurrent Neural Network: RNN*) y el enfoque de secuencia a secuencia de [Sutskever et al. (2014)]. Es completamente entrenable a partir de datos que incluyen el contexto anterior junto con las respuestas que se generarán. Muestran que el generador de contexto genera mejoras significativas sobre la línea base tanto en métricas automáticas como en una prueba de preferencia por parejas.

En el trabajo de [Li and Boyer (2016)] comentan que la comprensión del diálogo situado requiere la identificación de referentes en el entorno al que se refieren los participantes del diálogo. Este problema de resolución de referencia, a menudo en un entorno complejo con gran ambigüedad, es muy difícil. Proponen un enfoque que aborda estos desafíos al combinar la estructura semántica aprendida de las expresiones referentes con la historia del diálogo en un modelo basado en el *ranking*. Evalúan la nueva técnica en un corpus de diálogos humanos-humanos para la programación de computadoras. El trabajo propuesto hace un paso hacia el diálogo automatizado en entornos complejos de resolución de problemas.

En la investigación de [Sordoni et al. (2015)] presentan una arquitectura de red neuronal para la generación de respuestas que son a la vez sensibles al contexto y basadas en datos. Como tal, puede ser entrenada de extremo a extremo en cantidades masivas de datos de medios sociales. Comentan que, esta es la primera aplicación de un modelo de red neuronal a la generación de respuesta de dominio abierto, y creen que el presente trabajo sentará las bases para modelos más complejos por venir. Además, introducen una nueva técnica de extracción de múltiples referencias que muestra una promesa para la evaluación automatizada.

Así también [Vinyals and Le (2015)] presentan un enfoque simple del modelado conversacional. El modelo conversacional predice la oración siguiente dada la oración u oraciones anteriores en una conversación. La fortaleza de este modelo es

que puede ser entrenado de extremo a extremo y por tanto, requiere mucho menos reglas hechas a mano. Además, este modelo sencillo puede generar conversaciones sencillas dado un gran conjunto de datos de entrenamiento de conversación. Los resultados preliminares sugieren que, a pesar de optimizar la función objetivo-error, el modelo es capaz de conversar bien. Es capaz de extraer conocimiento de un conjunto de datos de dominio específico y de un conjunto de datos grande, ruidoso y de dominio general de subtítulos de películas. En un conjunto ruidoso de datos de transcripción de películas de dominio abierto, el modelo puede realizar formas sencillas de razonamiento de sentido común. También se encontró que la falta de consistencia es un modo de falla común de este modelo.

En el trabajo de [Yao et al. (2015)] se propone un enfoque basado en redes neuronales que modela los procesos de atención e intención. Consiste esencialmente en tres redes recurrentes. La red de codificadores es un modelo a nivel de palabra que representa oraciones origen. La red de intenciones es una red recurrente que modela la dinámica del proceso de intención. La red decodificadora es una red recurrente que produce respuestas a la entrada desde el origen. Es un modelo de lenguaje que depende de la intención y tiene un mecanismo de atención para atender a las palabras de origen en particular, al predecir un símbolo en la respuesta. El modelo está entrenado de extremo a extremo sin datos etiquetados. Los experimentos muestran que este modelo genera respuestas naturales a las entradas de los usuarios.

En el caso de [Wen et al. (2015)] presentan un generador de lenguaje estadístico basado en una estructura de memoria a corto y largo plazo (del inglés *Long Short Term Memory: LSTM*) controlada semánticamente. El generador de LSTM puede aprender de datos no alineados optimizando conjuntamente la planificación de oraciones y la realización superficial usando un criterio de entrenamiento de entropía cruzada simple, y la variación del lenguaje se puede lograr fácilmente mediante el muestreo de los candidatos de salida.

Una evaluación objetiva en dos diferentes dominios de prueba muestra que el método propuesto mejoró el rendimiento en comparación con los métodos anteriores. Los jueces humanos anotaron el sistema LSTM más alto en informatividad y

naturalidad y en general lo prefirieron a los otros sistemas.

Utilizando el mismo enfoque de estructura LSTM [Luan et al. (2016)] presentan un modelo de conversación que incorpora tanto el contexto como el rol del participante en las conversaciones de dos partes. El modelo conversacional puede funcionar como un modelo de lenguaje o un modelo de generación de lenguaje. Los experimentos realizados en el corpus de diálogo de Ubuntu<sup>10</sup> muestran que el modelo puede capturar la interacción de múltiples turnos entre los participantes. El método propuesto supera a un modelo LSTM tradicional medido por la perplejidad del modelo de lenguaje y la clasificación de respuestas. Las respuestas generadas muestran diferencias de características entre los dos roles de los participantes.

En el trabajo de [Li et al. (2016)] comentan que los modelos de redes neuronales de secuencia a secuencia de [Sutskever et al. (2014)] para la generación de respuestas conversacionales tienden a generar respuestas habituales y seguras (por ejemplo, “no sé”) independientemente de la entrada. Sugieren que la función objetivo tradicional, es decir, la probabilidad de salida (respuesta) dada la entrada (mensaje) no es adecuada para tareas de generación de respuesta. En su lugar, proponen utilizar máxima información mutua (del inglés *Maximum Mutual Information: MMI*) como la función objetivo en los modelos neuronales. Los resultados experimentales demuestran que los modelos de MMI propuestos producen respuestas más diversas, interesantes y apropiadas, produciendo ganancias sustanciales en las puntuaciones de BLEU<sup>11</sup> en dos conjuntos de datos conversacionales y en evaluaciones humanas.

En la investigación de [Su et al. (2016)] describen un enfoque de dos pasos para la gestión del diálogo en sistemas de diálogo oral orientados a tareas. Se propone un marco de red neuronal unificado para permitir al sistema aprender primero mediante supervisión a partir de un conjunto de datos de diálogo y luego mejorar continuamente su comportamiento a través del aprendizaje por refuerzo, todos utilizando algoritmos basados en gradiente en un solo modelo. Los experimentos demuestran la efectividad del modelo supervisado en la evaluación basada en el corpus, con simu-

---

<sup>10</sup>Corpus de diálogos de Ubuntu: 26 millones de turnos de diálogos naturales entre dos personas.

<sup>11</sup>BLEU: un método para la evaluación automática de la traducción automática (BLEU por BiLingual Evaluation Understudy).

lación de usuarios y con sujetos humanos remunerados. El uso del aprendizaje por refuerzo mejora aún más el rendimiento del modelo en ambos entornos interactivos, especialmente bajo condiciones de mayor ruido.

Los modelos neuronales de generación del diálogo ofrecen una gran promesa para generar respuestas para los agentes de conversación, pero tienden a ser miopes, prediciendo enunciados uno a la vez ignorando su influencia en los resultados futuros. Modelar la dirección futura de un diálogo es crucial para generar diálogos coherentes e interesantes, una necesidad que llevó a los modelos tradicionales de procesamiento de lenguaje natural (PLN; del inglés *Natural Language Processing: NLP*) de diálogo a basarse en el aprendizaje por refuerzo. Como en el trabajo de [Li et al. (2016)] que muestran cómo integrar estos objetivos, aplicando el aprendizaje por refuerzo profundo para modelar la recompensa futura en el diálogo chat-bot. El modelo simula diálogos entre dos agentes virtuales, utilizando métodos de políticas de gradiente para recompensar las secuencias que muestran tres propiedades conversacionales útiles: informatividad, coherencia y facilidad de respuesta. El modelo se evalúa sobre diversidad, longitud y jueces humanos, mostrando que el algoritmo propuesto genera respuestas más interactivas y logran fomentar una conversación más sostenida en la simulación del diálogo. Este trabajo marca un primer paso hacia el aprendizaje de un modelo neuronal de conversación basado en el éxito de los diálogos a largo plazo.

A manera de resumen, la Tabla 3.1 muestra una clasificación del estado de los modelos, autores, herramientas y técnicas para el dominio de datos empleados en la generación automática de diálogos que se ha discutido en este capítulo.

Como resultado de este análisis bibliográfico se puede observar que en ninguno de los trabajos revisados se utiliza modelo alguno de inferencia gramatical ni corpus en idioma español. Este hallazgo presenta un área de oportunidad y por lo tanto en este trabajo de tesis se propone un modelo de inferencia gramatical dentro del sistema de diálogo, utilizando el corpus en idioma español DIHANA. La propuesta metodológica de la aproximación al sistema de diálogo desarrollado en este trabajo de investigación se discute a detalle en el capítulo siguiente.

Tabla 3.1: Clasificación de los diferentes modelos utilizados en la generación automática de diálogos.

Modelo	Autor	Herramientas, técnicas y/o modelos	Dominio
Reglas Probabilísticas	Lison (2012)	Redes Bayesianas	Conjunto de datos de Wizard-of-Oz Twitter, WMT08, WMT09
	Ritter (2013)	SMT, GIZA++	
Procesos Estocásticos	Barlier et al., (2015)	MDP, POMDP y RL	100000,75000 diálogos
	Iijima y Kobayashi (2016)	POMDP, Q-learning	
	Lison (2013)	POMDP, Redes Bayesianas	
	Pietquin (2013)	MDP, RL	
	Misu et al., (2012)	MDP, POMDP, RL	
	Png (2011)	MDP, POMDP, BRL	
	Mahadik (2017)	POMDP, Redes Bayesianas	
Young y Thomson (2013)	POMDP y RL	Wizard-of-Oz y simulación de usuario Simulación del usuario Corpus (Museo de Ciencias de Boston) Corpus SACTH	
Redes Neuronales	Dušek y Jurčiček (2016)	LSTM y seq-seq	Crowdsourcing
	Sordani et al., (2015)	RNN, RLM	Conversaciones de Twitter
	Vinyals (2015)	RNN (LSTM) y seq2seq	OpenSubtitles (dominio abierto). Conjunto de datos de solución de problemas (dominio cerrado)
	Yao et al., (2015)	RNN, seq-seq	Diálogos de servicio de chat
	Wen et al., (2015)	LSTM	Ontologías (SF Restaurant, SF Hotel)
	Luan et al., (2016)	LSTM	Corpus de diálogo UBUNTU
	Li et al., (2016)	NN, seq-seq y MMI	OpenSubtitles
	Su et al., (2016)	NN, SL y RL	Conjunto de datos de Wizard-of-Oz (Amazon Mechanical Turk)
	Li et al., (2016)	LSTM y seq-seq, RL	OpenSubtitles
	Shang et al., (2015)	NN, SMT, GIZA++	Microblog similar a Twitter en China
	Sutskever (2014)	RNN, LSTM, seq-seq	WMT'14

# Capítulo 4

## Sistema de Diálogo propuesto

En este capítulo se describe la estructura general que se propone para un sistema de diálogos escritos del tipo pregunta-respuesta. Como parte de la estructura del sistema de diálogo propuesto se plantea una estrategia para poder identificar las entidades semánticas que aparecen en los diálogos, utilizando un Reconocedor de Entidades Nombradas (NER), el cual tiene como núcleo un clasificador de Campos Aleatorios Condicionales (CRF). También se plantea un modelo de inferencia gramatical de secuencias de componentes semánticos que representan a las preguntas y respuestas de los diálogos y mediante el cual es posible identificar las intenciones del usuario expresadas en lenguaje natural y generar plantillas de respuesta que posteriormente son alimentadas con información del mundo real a partir de una ontología creada de manera específica para el caso de estudio.

### 4.1. Estructura general del Sistema de Diálogo

Los sistemas de diálogo hablado o sistemas conversacionales son una tecnología concebida para facilitar la interacción natural mediante el habla, entre una persona y una computadora. Son básicamente una interfaz hombre-máquina capaz de reconocer y comprender una entrada hablada y reproducir una salida oral como respuesta. Los sistemas de diálogo consisten de una estructura modular en la que cada módulo se ocupa de determinadas tareas en interacción con todos los demás

módulos. Normalmente los módulos que conforman a un sistema de diálogo son: el reconocedor del habla, el módulo de comprensión del lenguaje, el gestor del diálogo y el módulo de generación de respuesta.

En este trabajo se propone una estructura general para el sistema de diálogo a desarrollar como se muestra en la Figura 4.1, el cual cuenta con los siguientes módulos: módulo de reconocimiento del lenguaje, módulo de inferencia gramatical, módulo de generación de plantilla de respuesta, módulo para la generación de respuesta, módulo de verificación y estructuración de la respuesta y un módulo para la gestión de la base de conocimientos.

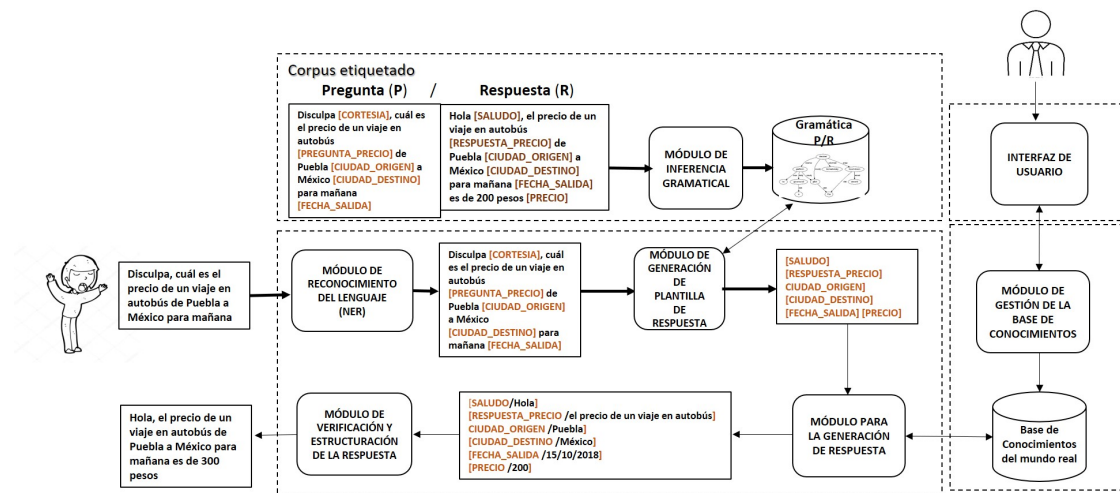


Figura 4.1: Estructura general del Sistema de Diálogo.

Con la finalidad de caracterizar de una manera mucho más simple la estructura del sistema de diálogo propuesto se han agrupado los seis módulos mencionados anteriormente en tres grandes módulos que se han denominado:

- Reconocimiento del lenguaje
- Inferencia gramatical
- Generación de la respuesta

La descripción de cada uno de estos grandes módulos sigue a continuación.

## 4.2. Reconocimiento del Lenguaje

En esta sección se describe el procedimiento para el reconocimiento del lenguaje natural en los diálogos que conforman el corpus de trabajo del sistema de diálogo propuesto. Se maneja el concepto de entidad semántica y el procedimiento para reconocer estas entidades semánticas a través del uso de reconocedores de entidades nombradas (NER) y posteriormente realizar el proceso de reconocimiento del lenguaje en la interpretación semántica (secuencia de entidades semánticas). Así mismo se describen las tres etapas que se requieren llevar a cabo para tener un modelo de NER entrenado y poderlo utilizar en el reconocimiento del lenguaje.

### 4.2.1. Reconocedor de Entidades Nombradas (NER)

En este trabajo se analizaron dos NER y su aplicación para el reconocimiento de las entidades semánticas que conforman los diálogos. Esta adaptación propuesta de manera innovadora en este trabajo de investigación apoya en el proceso de reconocimiento del lenguaje para la interpretación semántica (secuencia de entidades semánticas) de los diálogos. En la siguiente sección se describen las tres etapas que se deben llevar a cabo para obtener un modelo de NER entrenado y posteriormente utilizarlo en el reconocimiento de entidades semánticas.

#### 4.2.1.1. Descripción general del entrenamiento de un modelo NER

En el procesamiento de lenguaje natural, el reconocimiento de una entidad nombrada es una tarea de extracción de información que busca ubicar y clasificar elementos en texto en categorías predefinidas, como personas, organizaciones, lugares, expresiones de tiempo y cantidades entre otros.

Considere, por ejemplo, el siguiente texto:

*“Juan López compró 300 acciones de Acme Corp. en 2016”*

Al aplicarle un NER, es posible reconocer las entidades PERSONA, ORGANIZACIÓN y TIEMPO, con lo cual el NER arroja la siguiente cadena de salida:



“[Juan López] (*PERSONA*) compró 300 acciones de [Acme Corp.] (*ORGANIZACIÓN*) en [2016] (*TIEMPO*).

Como se puede deducir, este tipo de etiquetamiento permite reconocer cadenas de lenguaje natural y asociarlas a una etiqueta que generaliza el significado de las cadenas de texto para asociarlas a una categoría semántica. Es decir, una instancia se asocia con una categoría la cual puede ser utilizada para manejar de una manera más sencilla la interpretación del lenguaje.

El proceso de entrenamiento de un modelo NER consta de las siguientes etapas:

- **Primera etapa:** Esta etapa corresponde a la preparación y etiquetado de los datos de entrenamiento y prueba, la cual se describe gráficamente en la Figura 4.2, en donde inicialmente se tiene un corpus de entrada al cual se le debe aplicar un método de “*tokenización*” y posteriormente realizar el procedimiento de “*etiquetado*” del mismo, de acuerdo con los requerimientos de cada NER. En este sentido, una de las tareas más importantes del preprocesamiento de los diálogos es el etiquetado de éstos. Cabe mencionar que en este trabajo de investigación, el etiquetado del corpus del dominio elegido y conocido como DIHANA se realizó manualmente.

A manera de ejemplo, en la Figura 4.4 se presenta el etiquetado de una serie de intervenciones para el diálogo mostrado en la Figura 4.3, el cual está relacionado con preguntas y respuestas de un sistema de transporte.

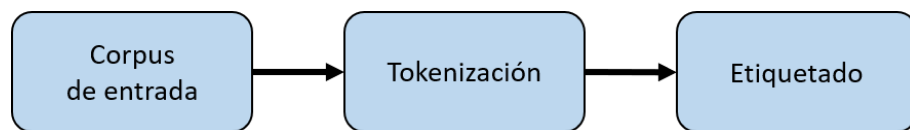


Figura 4.2: Preparación y etiquetado de los datos de entrenamiento y prueba para un NER.

- **Segunda etapa:** Esta etapa corresponde al entrenamiento del modelo NER, y normalmente se utiliza una parte del corpus etiquetado en la primera etapa. Esta parte del conjunto de datos se denomina corpus de entrenamiento (“*Training*”), y después de su ejecución arroja como resultado un modelo de

M1: Bienvenido al servicio de información de trenes . ¿ En qué puedo ayudarle? .  
 U1: Quisiera horarios de trenes para madrid el próximo viernes .  
 M2: ¿ Quiere horarios de trenes a madrid , desde zaragoza ? .  
 U2: Sí .  
 M3: Le consulto horarios de trenes con salida en zaragoza destino madrid saliendo el viernes , día 5 de marzo de 2004 . Un momento por favor .  
 M4: Hay varios trenes . El primero sale a las 2 y 48 de la madrugada y el último a las 7 y 10 de la tarde . ¿ Desea algo más ? .  
 U3: Sí , quisiera los horarios que llegan antes de la una .  
 M5: Lo siento . No he entendido bien . ¿ puede repetir ? .  
 U4: Quiero los horarios de trenes que llegan antes de la una de la tarde .  
 M6: Un momento por favor .  
 M7: Hay varios trenes . El primero sale a las 2 y 48 de la madrugada y llega a las 7 de la mañana y el último sale a las 9 de la mañana y llega a las 12 y 20 de la mañana . ¿ Desea algo más ? .  
 U5: No .  
 M8: Gracias por utilizar este servicio . Feliz viaje .

Figura 4.3: Ejemplo de un diálogo del corpus DIHANA donde se muestran las diferentes intervenciones del usuario y de la máquina.

M1: Bienvenido al servicio de información de trenes . ¿ En qué puedo ayudarle?  
 Bienvenido al servicio de información de trenes En qué puedo ayudarle:**apertura**  
 U1: Quisiera horarios de trenes para madrid el próximo viernes .  
 Quisiera:**consulta**  
 horarios de trenes:<**hora**>  
 para madrid:**ciudad\_destino**  
 el próximo viernes:**fecha**  
 M2: ¿ Quiere horarios de trenes a madrid , desde zaragoza ?  
 Quiere horarios de trenes:**confirmacion**  
 a madrid:**ciudad\_destino**  
 desde zaragoza:**ciudad\_origen**  
 U2: Sí .  
 U2: Sí .  
 U2: Sí .  
 U2: Sí .  
 M3: Le consulto horarios de trenes con salida en zaragoza destino madrid saliendo el viernes , día 5 de marzo de 2004 . Un momento por favor .  
 Le consulto horarios de trenes:**confirmacion**  
 con salida:**m\_salida**  
 en zaragoza:**ciudad\_origen**  
 destino madrid:**ciudad\_destino**  
 saliendo el viernes día 5 de marzo de 2004:**fecha**  
 Un momento por favor:**espera**

Figura 4.4: Ejemplo de diálogo etiquetado del corpus DIHANA.

NER entrenado, tal y como se describe en la Figura 4.5. Los procedimientos estándar para la selección del tamaño del corpus de entrenamiento sugieren el uso de un 80 % del global de datos. En el caso particular de este trabajo de tesis se han seguido estos lineamientos para garantizar concordancia con los procedimientos sugeridos en la literatura.

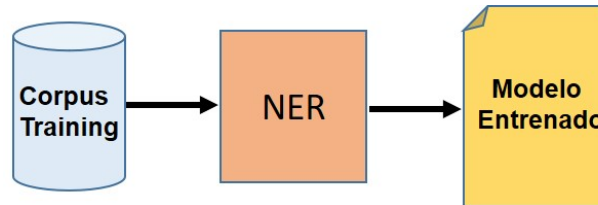


Figura 4.5: Etapa de entrenamiento de un NER.

- **Tercera etapa:** Corresponde a la evaluación del modelo obtenido en la etapa anterior utilizando el 20 % del corpus obtenido en la primer etapa, al cual se identifica como corpus de prueba (“*Test*”) y es aquí donde se ve el desempeño del modelo, verificando la precisión de las entidades semánticas identificadas, como se muestra en la Figura 4.6.

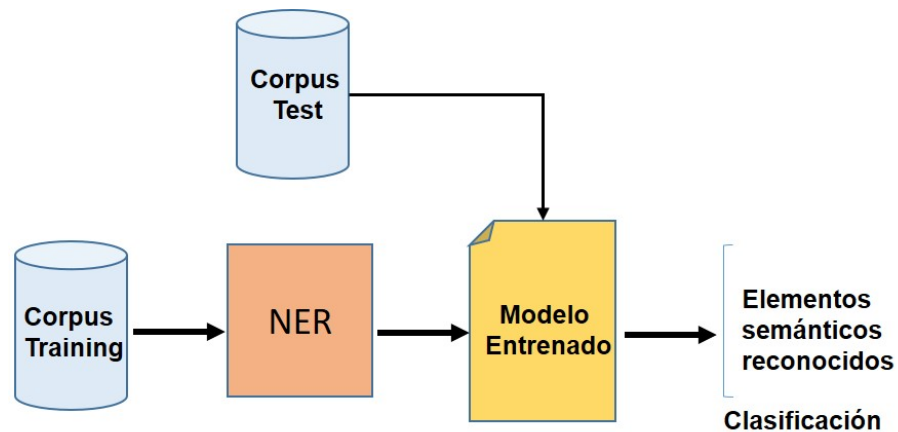


Figura 4.6: Etapa de evaluación de un NER.

Para el propósito de este trabajo de investigación fue necesario evaluar dos aproximaciones diferentes para el reconocimiento de entidades nombradas con la

finalidad de escoger el que presenta un mejor rendimiento y de esta manera reducir los errores por encadenamiento en los diferentes módulos. La descripción de los resultados experimentales para esta etapa se presentan a detalle en la Sección 5.3.1.

El reconocimiento del lenguaje se puede ver como un procedimiento que acepta una oración (cadenas de palabras) como entrada y produce una oración semántica (secuencia de entidades semánticas). Desde este punto de vista, el reconocimiento del lenguaje es un proceso de *transducción*, es decir transduce la oración de entrada en una oración semántica definida en un lenguaje semántico secuencial.

Ahora bien como se menciona en [Segarra et al. (2002)], si el lenguaje semántico es secuencial con el lenguaje de entrada, entonces se puede realizar una *segmentación* de la oración de entrada en una cantidad de intervalos que es igual al número de unidades semánticas en la oración semántica correspondiente. Es decir formalmente se define de la siguiente manera. Sea  $W$  el vocabulario de la tarea (conjunto de palabras), y sea  $V$  el alfabeto de las unidades semánticas, el conjunto de entrenamiento es un conjunto de pares  $(u, v)$  donde:

$$v = v_1v_2\dots v_n, \quad v_i \in V, \quad i = 1, \dots, n$$

$$u = u_1u_2\dots u_n, \quad u_i = w_{i_1}w_{i_2}\dots w_{i_{|u_i|}}, \quad w_{ij} \in W, \quad i = 1, \dots, n, \quad j = 1, \dots, |u_i|$$

Cada oración de entrada en  $W$  tiene un par  $(u, v)$  asociado, donde  $v$  es una secuencia de unidades semánticas y  $u$  es una secuencia de segmentos de palabras. En la Tabla 4.1 se muestra el resultado del reconocimiento del lenguaje para reconocer la oración “*Quisiera horarios de trenes para Madrid el próximo viernes*”.

Como se puede apreciar, el procedimiento de reconocimiento del lenguaje natural es un procedimiento de dos fases, la primera fase es el reconocimiento de entidades semánticas utilizando el NER de Stanford previamente entrenado con el corpus etiquetado como se mostro en la Figura 4.4, obteniendo las 9 entidades semánticas: *consulta <hora><hora><hora>ciudad\_destino ciudad\_destino fecha fecha fecha* y la segunda fase es el proceso de transducción, obteniendo finalmente la secuencia semántica: *consulta <hora>ciudad\_destino fecha* que representa semánticamente a la oración inicial *Quisiera horarios de trenes para Madrid el próximo viernes*.

Tabla 4.1: Reconocimiento del lenguaje natural aplicado a la oración “*Quisiera horarios de trenes para Madrid el próximo viernes*” que forma parte del corpus DIHANA.

<b>Oración de entrada a ser reconocida</b>
<i>Quisiera horarios de trenes para Madrid el próximo viernes</i>
<b>Segmentación de la oración etiquetada</b>
<i>Quisiera:consulta</i>
<i>horarios de trenes:&lt;hora&gt;</i>
<i>para Madrid:ciudad_destino</i>
<i>el proximo viernes:fecha</i>
<b>Oración desalida (9 entidades semánticas)</b>
<i>consulta &lt;hora&gt;&lt;hora&gt;&lt;hora&gt;ciudad_destino ciudad_destino fecha fecha fecha</i>
<b>Transducción, secuencia de entidades semánticas</b>
<i>consulta &lt;hora&gt;ciudad_destino fecha</i>

Siguiendo el mismo procedimiento de reconocimiento del lenguaje, en la Figura 4.7 se muestran las secuencias de entidades semánticas obtenidas después del reconocimiento del lenguaje al diálogo mostrado en la Figura 4.3.

```

apertura
consulta <hora> ciudad_destino fecha
confirmacion ciudad_destino ciudad_origen
<afirmacion>
confirmacion m_salida ciudad_origen ciudad_destino fecha espera
respuesta numero_relativo_orden m_salida hora numero_relativo_orden hora nueva_consulta
<afirmacion> consulta <hora_llegada> hora
no_entendido
consulta <hora_llegada> hora
espera
respuesta numero_relativo_orden m_salida hora m_llegada hora numero_relativo_orden m_salida hora m_llegada hora nueva_consulta
<negacion>
coletilla cierre

```

Figura 4.7: Secuencias de entidades semánticas obtenidas del reconocimiento del lenguaje del diálogo mostrado en la Figura 4.3.

Cabe resaltar la importancia que tiene este módulo de reconocimiento del lenguaje, ya que es el punto de partida para la generación de un modelo de inferencia gramatical que permitiera identificar las intenciones de un usuario basadas en expresiones generadas en lenguaje natural y a partir de estas proporcionar una respuesta coherente por parte del sistema de diálogo. Este módulo de inferencia gramatical se describe a detalle en la siguiente sección.

### 4.3. Inferencia Gramatical

En este capítulo se presenta el modelo de inferencia gramatical propuesto e implementado dentro del sistema de diálogo, cuyo principal objetivo es el de inferir un enunciado de respuesta en función de un enunciado de pregunta. Antes de describir el modelo de inferencia gramatical propuesto describiremos de manera general la idea de inferencia gramatical y algunos conceptos básicos.

#### 4.3.1. Conceptos básicos

Como precedente a la definición de un lenguaje regular, se define una gramática formal como una cuádrupla  $G = (N, \Sigma, P, A)$ , donde  $N$  es un conjunto de símbolos no terminales, es decir, se pueden sustituir mediante una regla de producción (se suelen representar en mayúsculas);  $\Sigma$  es el alfabeto, símbolos finales que no pueden ser sustituidos (en este caso, en minúsculas);  $P$  es un conjunto de producciones o reglas de transformación; y  $A \in N$  es el símbolo del axioma, es decir, el símbolo raíz. La diferencia entre los distintos tipos de gramáticas que define [Chomsky (1956)] se basa en las restricciones que tienen las producciones.

Los lenguajes regulares (LR) son el conjunto de los lenguajes formales más restrictivos de acuerdo con la jerarquía de [Chomsky (1956)]. Se describen a partir de las gramáticas de tipo 3, también conocidas como gramáticas regulares. Estas expresiones están formadas por una secuencia de caracteres que constituyen el patrón de todas las cadenas que pertenecen a un determinado lenguaje (ver ejemplo en Tabla 4.2, donde se introduce el lenguaje que contiene todas las cadenas con un número par de símbolos  $a$ : *Paridad*), ahí se describe el lenguaje regular con alfabeto  $\Sigma = \{a, b\}$  mediante su gramática regular derecha y su expresión regular equivalente: cadenas con número par de símbolos  $a$ . En la descripción de la gramática se utiliza el símbolo  $|$  como operador booleano OR con el fin de agrupar aquellas producciones con el mismo símbolo no terminal a la izquierda y el símbolo  $\lambda$  como representación de la cadena vacía. En la expresión regular equivalente se sigue la nomenclatura de [Kleene (1956)], donde el símbolo  $*$  representa 0 o más repeticiones del símbolo o

grupo al que aplica.

Las gramáticas regulares, son aquellas que reconocen la clase de los lenguajes regulares, son las más restrictivas: el lado izquierdo de una producción debe contener un símbolo no terminal, mientras que el lado derecho puede contener símbolos terminales y no terminales siguiendo siempre la misma estructura de linealidad, es decir, los símbolos terminales aparecen siempre al inicio (gramática regular derecha) o siempre al final (gramática regular izquierda).

Tabla 4.2: Ejemplo de la descripción de un lenguaje regular con alfabeto  $\Sigma = \{a, b\}$  mediante su gramática regular derecha y su expresión regular equivalente.

Gramática Regular	Expresión Regular
$A \rightarrow bA \mid aB \mid \lambda$	$b^*(ab^*ab^*)^*$
$B \rightarrow bB \mid aA$	

Así también, se sabe que las cadenas pertenecientes a un lenguaje regular dado pueden ser identificadas mediante un autómata finito, es decir, existe una equivalencia directa entre una gramática regular dada, su expresión regular correspondiente y el autómata finito que la identifica [Linz (2011)]. Un autómata finito determinista (AFD), o máquina de estados determinista, es un modelo computacional abstracto formado por un número finito de estados internos y una función de transición que, a partir de una entrada, produce una determinada salida.

Por definición un AFD se denota entonces como la quintupla  $A = (Q, \Sigma, \delta, q_0, F)$ , donde  $Q$  es el conjunto (finito) de estados,  $\Sigma$  es un alfabeto finito,  $\delta$  es una función de  $Q \times \Sigma \rightarrow Q$ ,  $q_0$  es el único estado inicial y  $F \subseteq Q$  es el conjunto de estados de aceptación (finales). El funcionamiento de los autómatas se basa en la función de transición  $\delta$ , que dado un estado  $q_{actual}$  y un símbolo del alfabeto, devuelve un estado  $q_{siguiente}$  al que se transita con dicho símbolo. Si después de procesar una cadena completa  $w \in \Sigma^*$  el estado interno del autómata es un estado final  $q_f \in F$ , entonces la cadena  $w$  pertenece al lenguaje, ya que cumple las producciones de la gramática formal. Además, se añade el concepto de determinismo cuando existe una única transición  $\delta_{q,s} \in \delta$  para cada estado  $q \in Q$  y símbolo  $s \in \Sigma$ . El conjunto de

todas las cadenas aceptadas por el autómata es un lenguaje regular.

En la Figura 4.8 se muestra el diagrama de estados que representa el AFD correspondiente al problema *Paridad*. En este diagrama se define completamente la quintupla del autómata: cada círculo identifica cada uno de los estados de  $Q$ ; las flechas entre los estados y los símbolos sobre ellas representan las transiciones  $\delta$  y el alfabeto  $\Sigma$ ; el estado  $q_0$  es el que tiene una flecha inicial; y los estados de aceptación  $F$  son aquellos con doble círculo.

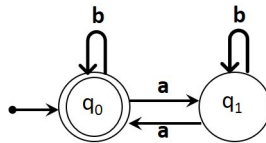


Figura 4.8: Diagrama de estados del AFD para el lenguaje regular *Paridad*.

### 4.3.2. Modelo de inferencia gramatical propuesto

Para este modelo de inferencia gramatical se utilizan las secuencias de entidades semánticas como las que se muestran en la Figura 4.7 que se obtienen a través del procedimiento de reconocimiento del lenguaje para cada uno de los diálogos a reconocer, como los que se muestran en la Figura 4.3 y esa es la idea del modelo de inferencia gramatical propuesto que, a través de las secuencias de entidades semánticas de un corpus de diálogos de entrada, se pueda inferir una gramática que reconozca dichos diálogos, en este caso la representación de la gramática es a través de un autómata finito determinista. Como ya se mencionó en la Sección 2.9, para el proceso de Inferencia Gramatical es necesario aprender de un conjunto de ejemplos positivos (conjunto de enunciados del corpus de diálogos de entrenamiento o aprendizaje) la gramática, para posteriormente extraer el autómata de estados finitos que reconoce el lenguaje asociado a los ejemplos de aprendizaje. Para el diseño de este modelo de inferencia gramatical se tomó como referencia la teoría de [Garcia and Vidal (1990)], en donde se consideran las clases conocidas de lenguajes  $k$ -comprobables ( $k$ -TLSS: Testable Languages in the Strict Sense). De manera



general, un  $k$ -TSSL está definido por un conjunto finito de subcadenas de longitud  $k$  que pueden aparecer en las cadenas del lenguaje. Los conceptos que están más o menos relacionados con los  $k$ -TLSS se han utilizado ampliamente en la teoría de la información. Los lenguajes estocásticos  $k$ -comprobables en sentido estricto están directamente relacionados con las fuentes de *Markov* de orden  $k$  y las frecuencias (probabilidades) de aparición de subcadenas de longitudes crecientes.

De una manera más formal se pueden definir a los lenguajes localmente comprobables de la siguiente manera.

### Lenguajes localmente comprobables

Suponga que  $Z_k = (\Sigma, I_k, F_k, T_k)$  es una cuádrupla donde  $\Sigma$  es un alfabeto finito,  $I_k, F_k \subseteq \cup_{i=1}^{k-1} \Sigma^i$  son dos conjuntos de segmentos inicial y final, respectivamente y  $T \subseteq \Sigma^k$  es un conjunto de segmentos prohibidos de longitud  $k$ . Un lenguaje  $k$ -comprobable en sentido estricto ( $k$ -TLSS) se define por la expresión regular

$$l(Z_k) = (I_k \Sigma^*) \cap (\Sigma^* F_k) - (\Sigma^* T_k \Sigma^*)$$

Por lo tanto, las cadenas en  $l(Z_k)$  pueden caracterizarse de la siguiente manera: comienzan con segmentos en  $I_k$  terminan con segmentos en  $F_k$  y no tienen ningún segmento de longitud  $k$  que esté en  $T_k$ .

### El $k$ -TLSS más pequeño que contiene una muestra positiva

Sea  $R$  un conjunto de aprendizaje (muestra positiva) y  $k \geq 1$ . Se puede asociar unívocamente una cuádrupla.  $Z_k(R) = (\Sigma(R), I_k(R), F_k(R), T_k(R))$  con  $R$  como sigue.

$$I_k(R) = \{u \mid uv \in R, |u| = k - 1, v \in \Sigma(R)^*\} \\ \cup \{x \in R \mid |x| < k - 1\}$$

(segmentos iniciales de longitud como máximo  $k - 1$  de las cadenas en  $R$ ).

$$F_k(R) = \{v \mid uv \in R, |v| = k - 1, u \in \Sigma(R)^*\}$$

$$\cup \{x \in R \mid |x| < k - 1\}$$

(segmentos finales de longitud como máximo  $k - 1$  de las cadenas en  $R$ ).

$$T_k(R) = \Sigma(R)^k - \{v \mid uvw \in R, |v| = k, u, w \in \Sigma(R)^*\}$$

(segmentos de longitud  $k$  que no aparecen en las cadenas en  $R$ ).

En consecuencia, un  $k$ -TLSS  $l(Z_k(R))$  se denotará como  $l_k(R)$ .

En función de lo expuesto anteriormente se describe el Algoritmo 1 de Inferencia Gramatical. Este algoritmo recibe como entrada al conjunto de cadenas de entrenamiento que son las secuencias de entidades semánticas obtenidas del módulo de reconocimiento del lenguaje, las cuales se denotan como  $S$  y como salida se obtiene un autómata finito determinista denotado como la quintupla  $A_k = (\Sigma, Q, \delta, q_0, Q_F)$ , donde  $\Sigma$  denota el alfabeto,  $Q$  el conjunto de estados,  $\delta$  las funciones de transición de un estado a otro,  $q_0$  el estado inicial y  $Q_F$  el conjunto de estados finales.

En función de lo expuesto anteriormente se describe el algoritmo de inferencia gramatical que recibe como entrada el conjunto de cadenas de entrenamiento que son las secuencias de entidades semánticas y obtiene como salida un autómata finito determinista. El algoritmo primero establece el alfabeto  $\Sigma$ , las transiciones  $\delta$ , también establece el conjunto de estados iniciales  $Q$  y finales  $Q_F$  y el estado inicial  $q_0$ . Posteriormente analiza todas y cada una de las cadenas de entrada, pertenecientes al conjunto de entrenamiento (cadenas de entrenamiento positivas), para cada cadena que analiza si es la primera vez que aparece crea una entrada en el alfabeto, así como un nuevo estado y su transición correspondiente si es necesario.

El algoritmo va creando los estados correspondientes y las funciones de transición necesarias. Es importante resaltar que el análisis de las cadenas lo realiza en función de la longitud de las mismas, y cada vez va incrementando el tamaño de las cadenas a analizar. También va creando los estados finales hasta analizar todas las cadenas de entrenamiento, obteniendo finalmente el autómata finito determinista inferido  $A_k = (\Sigma, Q, \delta, q_0, Q_F)$ .

Una vez diseñado e implementando el procedimiento de inferencia gramatical se

**Algoritmo 1:** Algoritmo de Inferencia Gramatical (k-TLSS)

---

```

Entrada   :  $k \in \mathbb{N}; S$  : Conjunto de cadenas          // Frases de entrenamiento
              positivas
Salida    :  $A_k = (\Sigma, Q, \delta, q_0, Q_F)$           // Autómata inferido
Variables :  $x, y$ : cadenas;  $q', q'', q$ : estados        // Representado como cadenas
              sobre  $\Sigma$ 
Inicializar:  $\Sigma = \delta = \phi; q_0 = \lambda; Q = \{q_0\}; Q_F = \phi$  //  $\lambda$  es la cadena vacía
for  $\forall x \in S$  do
   $q' = q_0$ 
  for  $i = 1 \dots |x|$  do
    if  $\exists q'' \mid (q', x, q'') \in \delta$  then  $q = q''$           // analiza utilizando la
    estructura actual
  else
     $\Sigma = \Sigma \cup \{x_i\}$                                 // Crea una nueva entrada al alfabeto
    // estado y/o transición, según sea necesario
     $y = q'x_i$ ; if  $|y| > k - 1$  then  $y = y_{2\dots|y|}$  endif
     $q = y$ 
     $Q = Q \cup \{q\}; \delta = \delta \cup \{(q'.x_i, q)\}$ 
    if  $i = |x|$  then  $Q_F = Q_F \cup \{q\}$  endif
  endif
   $q' = q$ 
end
end
 $A_k = (\Sigma, Q, \delta, q_0, Q_F)$ 

```

---

procede a generar la respuesta para la intervención del usuario. Este módulo se describe a detalle en la siguiente sección.

## 4.4. Generación de la respuesta

En primera instancia se implementa un módulo de generación de la plantilla de respuesta utilizando el AFD inferido en el módulo anterior. Este módulo inicia con la introducción de una oración (diálogo) del tipo pregunta, la cual está dada como una secuencia de entidades semánticas obtenida del reconocedor del lenguaje. Esta secuencia se analiza con el AFD inferido y procede a obtener o generar las posibles respuestas a la pregunta planteada, analizando las frecuencias de cada una de las entidades semánticas dentro del AFD. En el proceso de inferencia se van ge-

nerando marcas dentro del AFD para poder proporcionar la respuesta correcta y de esta manera generar la plantilla de respuesta a través de una secuencia de entidades semánticas inferidas del AFD. En la Figura 4.9 se muestra el esquema general del funcionamiento de este módulo, se puede apreciar en un proceso intermedio la generación de las etiquetas que marcan las posibles respuestas a la pregunta planteada (M1, M23, ..., M98), es decir, no hay una única respuesta ya que se tiene que buscar de entre las posibles respuestas aquella que sea mejor evaluada y de esta manera tener una respuesta coherente a la pregunta realizada y poder generar la plantilla de respuesta correcta por medio de secuencias de entidades semánticas inferidas en el AFD.

Algo importante en este módulo, es que en un diálogo (como una secuencia de entidades semánticas) no se hace referencia a una ciudad específica, ni hora o fecha específica y sin embargo se obtiene la respuesta coherente y correcta. Este procedimiento se repite con cada una de las preguntas que se realizan al sistema. En la Sección 5 se describen algunos ejemplos que muestran la manera en que se infieren estas respuestas para generar la plantilla de respuesta utilizando el AFD.

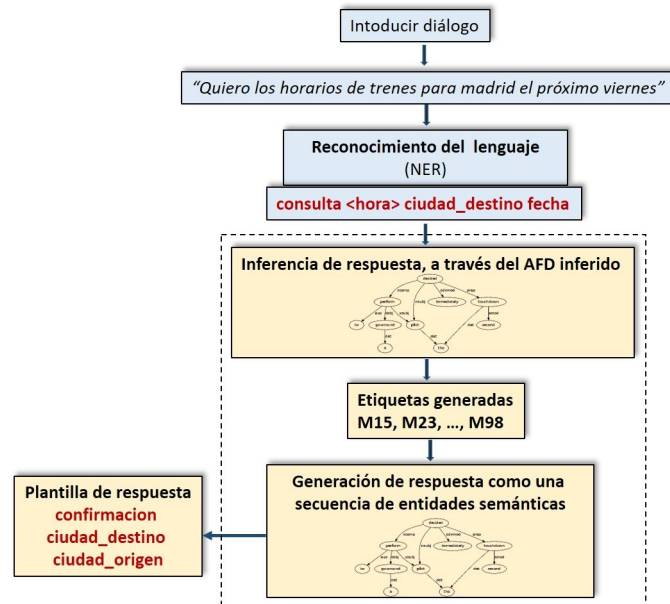


Figura 4.9: Esquema general del funcionamiento del módulo de generación de la plantilla de respuesta.

#### 4.4.1. Administración de la base de conocimiento

En esta sección se describe la base de conocimiento utilizada para generar la respuesta a través de la plantilla de respuesta descrita en la sección anterior.

Esta base de conocimiento contiene información del mundo real asociada con el dominio específico de aplicación y sirve para rellenar los campos que se encuentran en la plantilla de respuesta.

Por esta razón es necesario diseñar un modelo ontológico que tenga una correspondencia en términos de las entidades semánticas, pues será la única manera en que se puedan establecer las asociaciones unívocas entre las entidades semánticas de la plantilla de respuesta y aquellas que se encuentran expresadas en la ontología.

El procedimiento para llevar a cabo el llenado de la plantilla de respuesta comienza por una consulta traducida de la intervención original del usuario a un lenguaje SPARQL sobre la ontología diseñada para obtener una respuesta que contiene valores considerados instancias de un conjunto de clases ontológicas, que como anteriormente hemos mencionado, están asociadas a las entidades semánticas de interés para el dominio de aplicación.

Estas instancias pueden entonces ser asociadas con las entidades semánticas de la plantilla de respuesta y generar así una oración candidata de respuesta, la cuál tendrá que ser analizada posteriormente para garantizar que su estructura sintáctica y semántica sea la correcta.

El proceso de verificación de la oración de respuesta se discute en la siguiente sección.

#### 4.4.2. Verificación y estructura de la respuesta

En esta sección se explica cómo es que se obtiene finalmente la respuesta a una pregunta que se hace inicialmente en lenguaje natural de forma escrita, pasando por varias etapas hasta llegar a obtener la respuesta también en lenguaje natural de manera escrita. En la Figura 4.10 se muestra el módulo que realiza la construcción y verificación de la respuesta que finalmente se obtiene como resultado a la pregunta

que se hace.

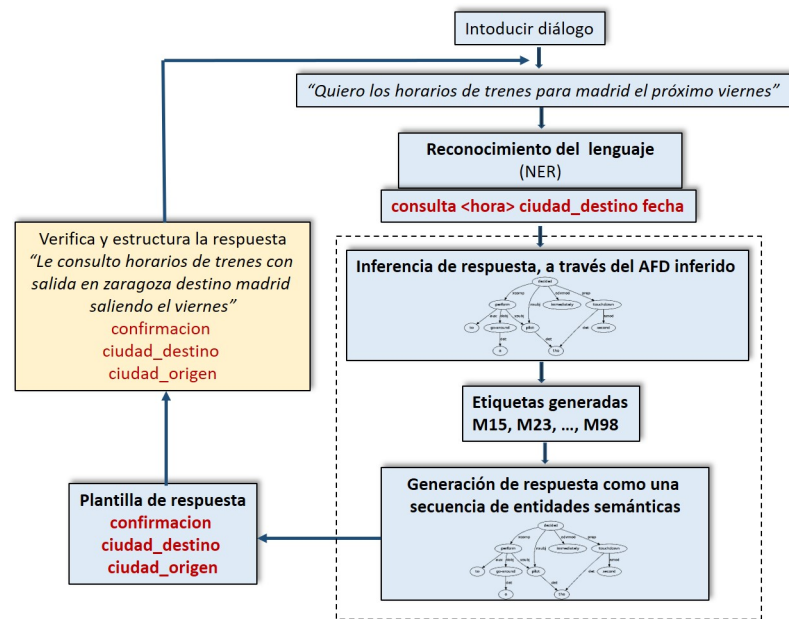


Figura 4.10: Esquema general del funcionamiento del Sistema de Diálogos en donde se muestra el módulo que construye y verifica la respuesta en lenguaje natural de manera escrita.

Para poder generar la respuesta se parte del resultado obtenido de la consulta a la ontología como se muestra en la Figura 5.14, los resultados que se tienen para cada una de estas partes que corresponden a las entidades semánticas y que representa la plantilla de respuesta se procesan y se va construyendo la respuesta de forma estructurada siguiendo exactamente la misma secuencia que nos da la plantilla de respuesta.

Para algunas etiquetas semánticas se tienen segmentos de oración asociadas a ellas, por ejemplo para la etiqueta *confirmacion* pueden ser: *Le consulto horarios de trenes* o *Le consulto precios de trenes* o *Le consulto tipos de trenes*, entonces para obtener el segmento de oración correcto se tiene el histórico de la pregunta que se hizo inicialmente y dependiendo del contexto de la pregunta se elige la mejor opción de acuerdo al contexto que puede ser *horario*, *precio* o *tipo*, etc. y de esta manera se va construyendo la respuesta .

En la Tabla 4.3 se muestran algunas etiquetas con sus correspondientes seg-

mentos de oración utilizados para construir y verificar que las respuestas estén bien construidas.

Tabla 4.3: Etiquetas y el valor asociado a cada etiqueta y su interpretación.

Etiqueta	Se refiere a	Valor/(es)
<i>afirmacion</i>	Una afirmación	“sí”
<i>negacion</i>	Una negación	“no”
<i>primero</i>	Frase para el primer tren que sale	“el primero”
<i>ultimo</i>	Frase pra el último tren que sale	“el último”
<i>pedir_fecha</i>	Frase para pedir la fecha	“¿que día desea salir?”
<i>confirmacion</i>	Frase para confirmación	“le consulto horarios” “le consulto precios” “le consulto tipos de trenes”
<i>apertura</i>	Frase de apertura del sistema	“Bienvenido al servicio de información de trenes”
<i>cierre</i>	Frase de despedida del sistema	“Gracias por utilizar este servicio”
<i>espera</i>	Frase de espera	“Un momento por favor”
<i>nueva_consulta</i>	Frase de nueva consulta	“la ida”
<i>viaje_redondo</i>	Frase del tipo de viaje redondo	“ida y vuelta”
<i>tip_tipo_viaje</i>	Frase de sugerencia para el tipo de viaje	“Recuerde que el billete de ida y vuelta siempre supone un ahorro de 20 por ciento”
<i>cortesia</i>	Frase para un saludo	“Hola” “Hola que tal” “Que tal” “Buenos días”
<i>coletilla</i>	Frase de despedida o agradecimiento	“Gracias” “Nada más gracias” “Muchas gracias” “Nada más muchas gracias” “Adios”

Una parte importante que permite verificar que la estructura sintáctica y semántica ha sido construida correctamente es el uso de una tecnología basada en modelos de lenguaje. La construcción de un modelo estadístico para modelar la probabilidad de una oración para el dominio de estudio se realiza mediante una aproximación basada en bigramas usando el conjunto de datos de entrenamiento.

De esta manera, una vez que se tiene una secuencia de palabras para una respuesta candidata se calcula la probabilidad de esta secuencia usando el modelo de lenguaje calculado mediante el corpus de entrenamiento y se determina su probabilidad de ocurrencia para el lenguaje asociado con el conjunto de diálogos de

entrenamiento.

Este último módulo permite generar oraciones mucho más naturales y que son percibidas por el usuario de una manera satisfactoria.

La descripción de las pruebas y los resultados obtenidos se presentan en el siguiente capítulo.



# Capítulo 5

## Pruebas y resultados

Una vez descrito el diseño de la metodología propuesta para la generación automática de diálogos en Español en este trabajo de tesis doctoral, en este Capítulo se presentan los resultados obtenidos bajo una serie de experimentos que tienen como objetivo valorar la calidad de la propuesta metodológica.

### 5.1. Motivación de las pruebas

En este Capítulo se presentan una serie de pruebas que tienen como objetivo determinar el rendimiento que tiene el sistema de diálogo propuesto. Valorar la calidad global de la propuesta puede ser algo complejo, pues requiere identificar el comportamiento de todos los módulos que componen al sistema. Por esta razón, esta evaluación se realiza en base a los grandes módulos que conforman el sistema propuesto. Para cada uno de ellos se describe la manera en que se propone realizar la prueba y se discuten enseguida los resultados obtenidos.

Es importante mencionar que para este trabajo de tesis doctoral, se utilizó un corpus particular en el dominio de sistemas de transporte (Trenes) para realizar los procesos de entrenamiento y prueba de la metodología propuesta. Por esta razón, en la siguiente sección se describe a detalle el conjunto de datos usado, y posteriormente se presentan las pruebas y los resultados que permiten evaluar el rendimiento de la propuesta de tesis doctoral.

## 5.2. Corpus DIHANA

El corpus en español DIHANA [Benedí et al. (2006)] está compuesto por 900 diálogos sobre un sistema de información de consultas telefónicas sobre horarios y precios de trenes. Fue adquirido por 225 hablantes diferentes (153 hombres y 72 mujeres). Hay 6,280 turnos de usuario y 9,133 turnos del sistema. El tamaño del vocabulario es de 823 palabras. La cantidad total de señal de voz fue de aproximadamente cinco horas y media.

La adquisición del corpus DIHANA se llevó a cabo por medio de un prototipo inicial, utilizando la técnica de Mago de Oz (WoZ). Esta adquisición solo se restringió a nivel semántico (es decir, los diálogos adquiridos están relacionados con un dominio de tareas específico) y no se restringió a nivel léxico y sintáctico (habla espontánea). En este proceso de adquisición, el control semántico fue proporcionado por la definición de escenarios que el usuario tenía que cumplir y por la estrategia WoZ, que define el comportamiento del sistema de adquisición. La Tabla 5.1 muestra las principales características del corpus DIHANA.

Tabla 5.1: Principales características del corpus DIHANA.

Característica	Valor
<i>Diálogos</i>	900
<i>Intervenciones promedio</i>	7-8
<i>Hablantes</i>	225 (153 H, 72 M)
<i>Turnos usuario</i>	6,280
<i>Turnos máquina</i>	9,133
<i>Vocabulario</i>	823

En la Figura 5.1 se muestra una conversación (un diálogo) perteneciente al corpus DIHANA. En este diálogo, se pueden observar las diferentes intervenciones por parte de la máquina (M) y del usuario (U). Tal y como se puede observar en el ejemplo mostrado, los turnos de la máquina (M) pueden tener más de una intervención. Lo cual no es el caso para las intervenciones del usuario (U).

Todos los diálogos forman parte de un archivo de texto, que como se puede observar es texto plano, el cual se debe de preparar (preprocesar) para que pueda ser utilizado en el sistema de diálogo.

M1: Bienvenido al servicio de información de trenes . ¿ En qué puedo ayudarle? .  
U1: Quisiera horarios de trenes para madrid el próximo viernes .  
M2: ¿ Quiere horarios de trenes a madrid , desde zaragoza ? .  
U2: Sí .  
M3: Le consulto horarios de trenes con salida en zaragoza destino madrid saliendo el viernes , día 5 de marzo de 2004 . Un momento por favor .  
M4: Hay varios trenes . El primero sale a las 2 y 48 de la madrugada y el último a las 7 y 10 de la tarde . ¿ Desea algo más ? .  
U3: Sí , quisiera los horarios que llegan antes de la una .  
M5: Lo siento . No he entendido bien . ¿ puede repetir ? .  
U4: Quiero los horarios de trenes que llegan antes de la una de la tarde .  
M6: Un momento por favor .  
M7: Hay varios trenes . El primero sale a las 2 y 48 de la madrugada y llega a las 7 de la mañana y el último sale a las 9 de la mañana y llega a las 12 y 20 de la mañana. ¿ Desea algo más ? .  
U5: No .  
M8: Gracias por utilizar este servicio . Feliz viaje .

Figura 5.1: Ejemplo de un diálogo del corpus DIHANA; se muestran las diferentes intervenciones del usuario y de la máquina.

Además del preprocesamiento hecho a este corpus, se consideró un proceso de etiquetado a nivel semántico para poder llevar a cabo las actividades de las pruebas previstas en este Capítulo. Las etiquetas semánticas representan una acción de una secuencia de palabras de la oración en el diálogo, también denominados como “actos del diálogo”.

Adicionalmente, y durante los procesos asociados con la metodología presente en el sistema de diálogo propuesto se toma ventaja de la estructura sintáctica para llevar a cabo las tareas asociadas con la inferencia de la gramática del lenguaje. También se utiliza el corpus para calcular las probabilidades de asociación de  $n$ -gramas de longitud 2 (bigramas) con la intención de verificar la estructura final de la respuesta.

A continuación se presenta la validación de los módulos del sistema, presentando primeramente la descripción del experimento y posteriormente la evaluación misma de los experimentos.

### 5.3. Validación de los módulos del sistema

El objetivo de los experimentos que se presentan a continuación es analizar el desempeño de los principales módulos del sistema. Es importante recordar que la metodología propuesta considera una categorización basada en tres módulos:

1. Módulo de reconocimiento del lenguaje
2. Módulo de inferencia gramatical
3. Módulo de generación de respuesta

Antes de presentar las pruebas y resultados de cada uno de estos módulos, es importante indicar que para el proceso de validación se utilizó en todos los casos una distribución del 80% del corpus de diálogos DIHANA<sup>12</sup> para el proceso de entrenamiento y el 20% restante para la fase de prueba.

#### 5.3.1. Módulo de reconocimiento del lenguaje

Como ya se mencionó en la Sección 4.2, se desarrolló e implementó una técnica basada en el reconocimiento de entidades nombradas para transformar textos planos (correspondientes a las intervenciones de diálogos) a secuencias de entidades semánticas que puedan ser perfectamente entendibles por una gramática computacional.

La descripción de la prueba para este módulo sigue a continuación.

##### 5.3.1.1. Descripción de la prueba

Este módulo está destinado a interpretar una oración (secuencia de palabras) escrita en lenguaje natural, y que corresponde a una intervención en el diálogo, para transformarla en una secuencia de componentes semánticos que puedan interpretarse más fácilmente por métodos computacionales.

---

<sup>12</sup>Como se mencionó en la Sección 5.2, el corpus DIHANA consta de 900 conversaciones y cada conversación tiene un promedio de 7 u 8 intervenciones entre el usuario y el sistema

Tal y como se comentón en la Sección 4.2, se tomó una aproximación basada en las técnicas de reconocimiento de entidades nombradas, por lo que el procedimiento de evaluación consiste en calcular la exactitud del sistema para identificar de manera automática las entidades semánticas que se encuentren presentes en un texto obtenido del corpus de prueba.

Si bien el corpus está escrito en Español, el etiquetado semántico fue considerado en Inglés, por motivos de dar una mejor interpretación a las lecturas de los artículos de investigación derivados de los experimentos llevados a cabo. En la Figura 5.2 se muestra un ejemplo de diálogos y su correspondiente reconocimiento del lenguaje, ahora en secuencias de entidades semánticas en idioma inglés.

<p>M1: Bienvenido al servicio de información de trenes . ¿ En qué puedo ayudarle? .          Bienvenido al servicio de información de trenes:opening          En qué puedo ayudarle:ask_for_query          U1: Me gustaría saber los horarios de los trenes que hay hacia Pamplona en clase turista .          Me gustaría saber:question          los horarios de los trenes:schedules          que hay hacia Pamplona:destination_city          en clase turista:trip_class          M2: ¿ Qué día desea salir? .          Qué día desea salir:ask_for_date          U2: El próximo sábado .          El próximo sábado:date          M3: Le consulto horarios de trenes con salida en valencia destino pamplona saliendo el sábado ,          día 1 de mayo de 2004 en clase turista . Un momento por favor .          Le consulto horarios de trenes:check_schedules          con salida en valencia:origin_city          destino pamplona:destination_city          saliendo el día 1 de mayo de 2004:departure_date          en clase turista:trip_class          Un momento por favor:wait          M4: El primero sale a las 7 y 3 de la mañana y llega a la una y 35 de la tarde y el último sale a las          10 y 3 de la noche y llega a las 7 de la mañana . Desea algo más? .          El primero:first          sale a las 7 y 3 de la mañana:departure_time          y llega a la una y 35 de la tarde:arrival_time          y el último:last          sale a las 10 y 3 de la noche:departure_time          y llega a las 7 de la mañana:arrival_time          Desea algo más:new_query</p>	<p>M_1 opening ask_for_query U_1          U_1 question schedules destination_city trip_class M_2          M_2 ask_for_date U_2          U_2 date M_3?          M_3? check_schedules origin_city destination_city departure_date          trip_class wait U_3!          U_3! M_4          M_4 first departure_time arrival_time last departure_time2          arrival_time2 new_query U_4          U_4 question train_type M_5?          M_5? check_prices origin_city destination_city departure_date          trip_class train_type wait U_5!          U_5! M_6          M_6 train_type trip_class price one_way_type tip_trip_type new_query          U_6          U_6 question destination_city M_7          M_7 ask_for_date U_7          U_7 date M_8?          M_8? check_schedules origin_city destination_city departure_date wait          U_8!          U_8! M_9          M_9 first departure_time last departure_time2 new_query U_9          U_9 negation tag M_10          M_10 closing</p>
--	--

Figura 5.2: Ejemplo de una conversación etiquetada en inglés y la secuencia de entidades semánticas reconocidas por el módulo de reconocimiento de lenguaje.

En la siguiente sección de presentan los resultados de evaluación para el módulo de reconocimiento del lenguaje.

### 5.3.1.2. Evaluación de la prueba

En este proceso de validación se utilizó una distribución del 80 % del corpus de diálogos DIHANA para el proceso de entrenamiento de las entidades nombradas y el 20 % restante para la fase de prueba.

Después de haber realizado las pruebas del modelo de reconocimiento del lenguaje, se procedió a validarlo para verificar el rendimiento de la propuesta planteada para el reconocimiento de las etiquetas semánticas. Los resultados de reconocimiento de las etiquetas semánticas son favorables, tal como se reportaron en [Vázquez et al. (2018)].

Con el propósito de condensar la evaluación de este módulo, a continuación se presenta un resumen de dichos resultados.

Los dos NER que se analizaron en este trabajo fueron: el NER de [The Stanford Natural Language Processing Group (2011)] y el NER de [Yuan (2014)]; estos dos NER utilizan el modelo de secuencia de Campos Aleatorios Condicionales (CRF), para el proceso de clasificación.

En ambos modelos se utilizaron los mismos corpus de entrenamiento y de prueba del corpus DIHANA de [Benedí et al. (2006)].

Los resultados globales de la exactitud obtenida en los dos NER se muestran en la Tabla 5.2. Esta medida de evaluación se calcula como el número de tokens de un tipo dado que el modelo identificó correctamente, entre el número total de tokens que el modelo predijo que era de ese tipo.

Tabla 5.2: Exactitud obtenida en la identificación de tokens.

NER	No. Tokens	Exactitud
Stanford	10,485	0.84
Xingdi-(Eric) Yuan	10,485	0.89

Se puede observar que el reconocedor de entidades nombradas de Xingdi (Eric) Yuan obtuvo el mejor resultado global. El resultado global muestra un promedio general, pero es muy importante analizar este resultado obtenido para cada una de las clases semánticas clasificadas. Por esta razón, en la Tabla 5.3, se han incluido los resultados para cada una de las entidades semánticas; los cuales se analizaron y se describen a continuación.

Las etiquetas semánticas “<hora\_llegada>”, “fecha”, “ciudad\_destino”, “<afirmación>”, “hora” y “ciudad\_origen” son las que han obtenido el mayor grado de

Tabla 5.3: Resultados obtenidos de la evaluación de los dos NER.

Etiquetas semánticas	Frecuencia	Frecuencia de	Exactitud NER	
	Entrenamiento	Prueba	Stanford %	Xingdi (Eric) Yuan %
<hora_llegada>	73	73	94.12	100.00
fecha	1678	1655	92.63	98.63
ciudad_destino	611	601	97.89	98.36
<afirmacion>	541	527	97.30	97.41
hora	978	949	87.50	97.03
ciudad_origen	323	310	98.72	95.98
<precio>	917	866	74.29	94.44
<hora>	618	578	81.52	93.53
consulta	1935	1758	80.03	90.85
<negacion>	344	302	85.61	87.79
numero_relativo_orden	24	21	62.50	87.50
m_salida	186	159	84.92	85.48
cortesía	253	215	85.37	84.98
m_llegada	136	115	69.62	84.56
coletilla	225	188	76.12	83.56
tipo_viaje	395	328	50.67	83.04
<tipo_tren>	65	52	58.33	80.00
tipo_tren	246	187	64.58	76.02
clase_billete	94	70	70.00	74.47
ciudad	44	30	77.78	68.18
<hora_salida>	38	20	85.71	52.63
precio	15	6	66.67	40.00
O	471	171	34.15	36.31
not	12	2	0.00	16.67
<duracion>	8	0	0.00	0.00
<no_entendido>	22	0	0.00	0.00
nombre_atributo	4	0	0.00	0.00

exactitud durante el proceso de identificación automática con un valor que supera el 95 %. Este resultado se encuentra fundamentado en dos cosas: primeramente, en la cantidad de datos usados durante la fase de entrenamiento, pero también a la inherente naturaleza de las palabras del lenguaje natural asociadas a dichas entidades semánticas, pues existe cierta uniformidad para expresar, horarios y fechas, y en cuanto a los nombres de las ciudades, se considera que el número es ciertamente limitado, lo cual facilita el proceso de identificación. Estos resultados fueron publicados en [Vázquez et al. (2018)].

Después de analizar el desempeño y la flexibilidad de uso de cada NER en los experimentos anteriormente mencionados, se optó por utilizar el NER de Stanford para el procedimiento de reconocimiento del lenguaje de este trabajo de tesis doctoral, debido a que presentó los mejores valores de exactitud.

### 5.3.2. Módulo de inferencia gramatical

Recordemos que el módulo de inferencia gramatical tiene como propósito el reconocimiento de cadenas asociadas a las intervenciones del usuario dentro de la gramática inferida, para posteriormente, colocar el sistema en modo generativo con la finalidad de proporcionar, como salida, una plantilla de respuesta en un formato de secuencias de entidades semánticas. De esta manera, en esta sección se presentan las pruebas y procesos de validación para el módulo que se ha denominado como de inferencia gramatical.

#### 5.3.2.1. Descripción de la prueba

Este módulo toma como entrada las secuencias de entidades semánticas generadas en el módulo anterior (que corresponde a un procedimiento de reconocimiento del lenguaje) y genera como salida una plantilla de respuesta conformada por entidades semánticas organizadas secuencialmente, siguiendo una estructura sintáctica.

Para la prueba y validación de este módulo, se utilizaron secuencias de diálogos etiquetados semánticamente. A manera de ejemplo, a continuación se presenta un diálogo conformado por dos intervenciones del usuario y tres por parte de la computadora, y una descripción detallada de la manera en que se procesan los diálogos para inferir la gramática y a partir de ahí generar la denominada plantilla de respuesta:

**Intervención 1 (Usuario):** *Me gustaría saber los horarios de los trenes que hay hacia Pamplona en clase turista*

**Intervención 2 (Máquina):** *Que día desea salir*

**Intervención 3 (Usuario):** *El próximo sábado*

**Intervención 4 (Máquina):** *Le consulto horarios de trenes destino pamplona saliendo el día 1 de mayo de 2004 en clase turista Un momento por favor*



**Intervención 5 (Máquina):** *El primero sale a las 7 y 3 de la mañana y llega a la una y 35 de la tarde y el ultimo sale a las 3 y 10 de la noche y llega a las 7 de la mañana Desea algo mas*

Las correspondientes secuencias de entidades semánticas generadas por el modelo de reconocimiento del lenguaje que están asociadas a las intervenciones que se acaban de presentar son las siguientes:

**Intervención 1 (Usuario):** *“question schedules destination\_city trip\_class”*

**Intervención 2 (Máquina):** *“ask\_for\_date”*

**Intervención 3 (Usuario):** *“date”*

**Intervención 4 (Máquina):** *“check\_schedules origin\_city destination\_date trip\_class wait”*

**Intervención 5 (Máquina):** *“first departure\_time arrival\_time last departure\_time2 arrival\_time2 new query”*

Como se puede observar, cada una de las intervenciones ha sido transformada de una secuencia de palabras escritas en lenguaje natural, a entidades semánticas que generalizan el contenido y facilitan la interpretación global de la intención del usuario.

Las intervenciones formadas por secuencias de entidades semánticas se introducen en el algoritmo de inferencia gramatical para crear un grafo que mantiene en memoria la información inferida, en este caso, nodos y arcos que indican la probabilidad de pasar de un estado A a un estado B mediante una cadena (etiqueta semántica).

La implementación del algoritmo de inferencia gramatical se realizó en lenguaje “C”. Se genera como salida un archivo de texto en un formato estándar, el cual contiene la información referente al autómata inferido. Las variables que se consideran en este archivo de textos son las siguientes:

$S_i$  : Identificador del estado del que parte el arco

$S_f$  : Identificador del estado al cual llega el arco

$I_S$  : Cadena de entrada asociada al arco

$P$  : Probabilidad asociada al arco

$O_S$  : cadena de caracteres de salida asociada al arco

Tomando como base este archivo de texto y con la finalidad de poder visualizar gráficamente el autómata inferido, se transforma el archivo en formato estándar a un archivo en formato *json*, usando un algoritmo de conversión específico.

Los procedimientos de reconocimiento de cadenas aceptadas por la gramática inferida, así como la generación de la plantilla de respuesta extraída del mismo grafo inferido ha sido implementado también en una librería programada en Python.

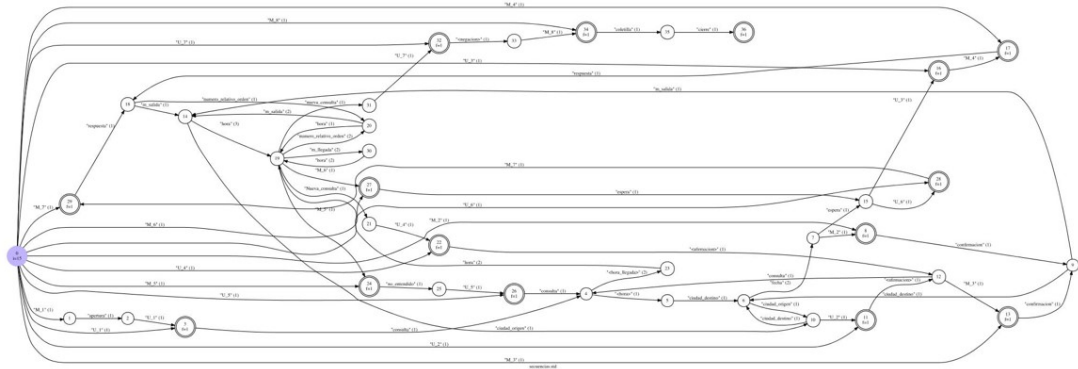


Figura 5.3: Ejemplo de un Autómata Inferido con el modelo propuesto.

El autómata que se muestra en la Figura 5.3 es un ejemplo que el lector puede observar para clarificar el proceso de construcción de la gramática inferida. Para este autómata, se utilizaron únicamente las intervenciones del diálogo anteriormente mostrado. El autómata completo fue entrenado con el 80 % de las conversaciones del corpus DIHANA; como puede inferirse, este autómata no es claramente visible en una figura debido a la cantidad de estados y transiciones que contiene, sin embargo, se adjunta (previa advertencia al lector) en la Figura 5.4 con la finalidad de mostrar la complejidad de esa estructura en forma de grafo.

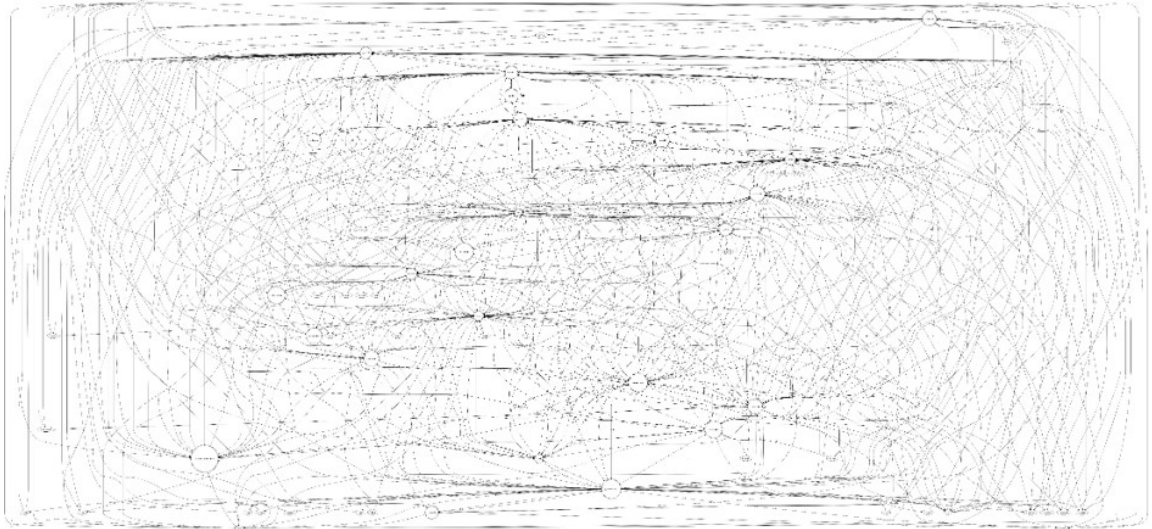


Figura 5.4: Autómata Inferido con el algoritmo de Inferencia, utilizando el 80 % de conversaciones del corpus DIHANA.

#### 5.3.2.2. Evaluación de la prueba

Se remarca que para la validación de este módulo se procedió igual que en el módulo de reconocimiento del lenguaje, utilizando una distribución del 80 % del corpus de diálogos DIHANA para el proceso de entrenamiento y el 20 % restante para la fase de prueba.

Se usan las secuencias de entidades semánticas que se encuentran en los diálogos del corpus de prueba para validar el módulo en un esquema de validación cruzada a 10 pliegues, observándose que se obtiene en promedio un 96.7 % de precisión para el rendimiento del módulo de inferencia gramtical. Este resultado ya fue reportado en [Vázquez et al. (2020a)]. Es importante mencionar que este valor de precisión se refiere exclusivamente al procedimiento de identificación de cadenas en la gramática inferida. En la Tabla 5.4 se muestra el resultado de la validación cruzada a 10 pliegues en el reconocimiento de secuencias de entidades semánticas de los diálogos del corpus de prueba.

Tabla 5.4: Resumen de la validación cruzada a 10 pliegues para la prueba de identificación de secuencias de entidades semánticas en la gramática inferida.

Iteración	Entrenamiento	Prueba	Aceptado	Rechazado	Precisión
1	3618	402	374	28	93.03
2	3606	414	400	14	96.61
3	3564	456	437	19	95.83
4	3615	405	393	12	97.03
5	3616	404	395	9	97.77
6	3632	388	369	19	95.10
7	3614	406	395	11	97.29
8	3639	401	398	3	99.25
9	3641	379	371	8	97.88
10	3621	399	390	9	97.74

### 5.3.3. Generación de la plantilla de respuesta

Con la intención de mostrar el funcionamiento para este submódulo, es preciso retomar paso por paso el procedimiento global de inferencia gramatical. Primero se consideran las secuencias semánticas generadas por el modelo de reconocimiento del lenguaje para entrenar el modelo de inferencia gramatical, e inferir el autómata que se muestra en la Figura 5.5.

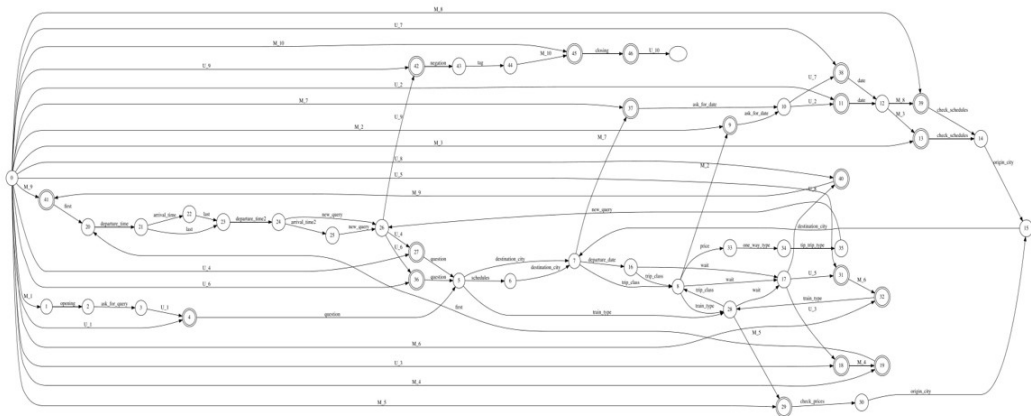


Figura 5.5: Autómata inferido con el modelo de inferencia gramatical entrenado con secuencias de entidades semánticas modeladas en el idioma Inglés .

Una vez obtenido el modelo de inferencia gramatical entrenado con las nuevas secuencias semánticas en inglés se procede a probar el módulo de generación de

plantilla de respuesta siguiendo la secuencia de las intervenciones indicadas anteriormente.

En la Figura 5.6 se presenta el proceso de reconocimiento de la secuencia de entidades semánticas dentro del autómata: “*question schedules destination\_city trip\_class*” cuando se hace la pregunta: “*Me gustaría saber los horarios que hay hacia Pamplona en clase turista*”. A partir del estado cuatro (4) que aparece en la parte lateral izquierda, se inicia el reconocimiento de la secuencia semántica siguiendo las aristas en color oscuro y los correspondientes nodos también en color oscuro (5, 6, 7 y 8) hasta llegar a un estado de posible respuesta (estado final). En este caso, el modelo infiere la respuesta “*ask\_for\_date*” como se muestra en la Figura 5.7, lo cual significa semánticamente que habría que colocar una respuesta coherente del tipo “*Qué día desea salir?*” o alguna oración similar escrita en lenguaje natural.

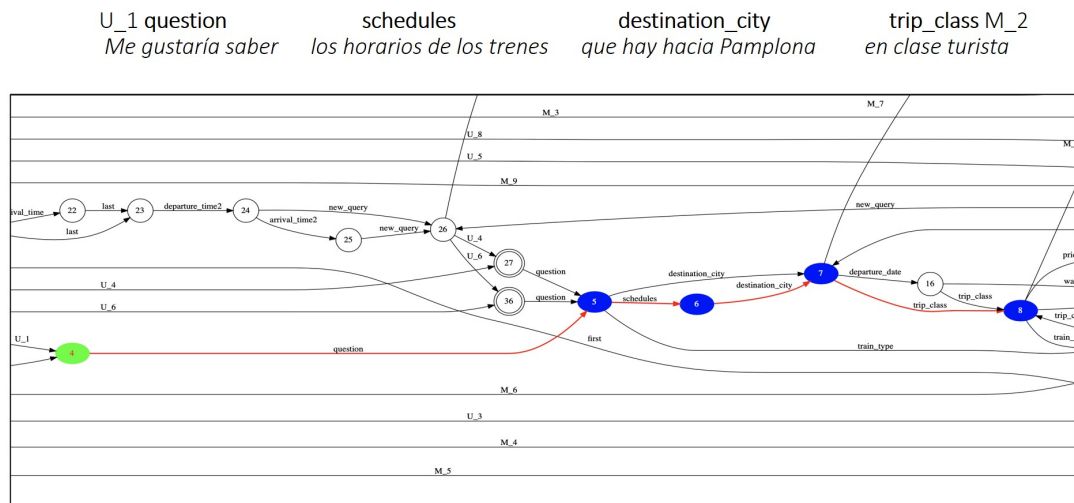


Figura 5.6: Reconocimiento de la secuencia de entidades semánticas: “*question schedules destination\_city trip\_class*”, en el autómata inferido .

A partir de la plantilla de respuesta generada, se espera que el usuario continúe con una intervención del diálogo del tipo mostrado en la Figura 5.8, en donde se muestra el reconocimiento de la etiqueta semántica “*date*” que corresponde a la siguiente intervención del usuario en el diálogo “*El próximo sábado*”. Nuevamente en la Figura 5.9 se muestra en el autómata la respuesta inferida a esta nueva intervención: “*check\_schedules origin\_city destination\_date trip\_class wait*”, que corresponde

a una respuesta generada como la siguiente: “*le consulto horarios de trenes con salida en valencia destino pamplona saliendo el día 1 de mayo de 2004 en clase turista. Un momento por favor*”. El modelo de inferencia gramatical también infiere cuando hay más de una intervención por parte del usuario o del sistema y en este caso en la Figura 5.10 se muestra una segunda intervención por parte del sistema infiriendo la respuesta: “*first departure\_time arrival\_time last departure\_time2 arrival\_time2 new query*” que corresponde al diálogo: “*El primero sale a las 7 y 3 de la mañana y llega a la una y 35 de la tarde y el lunes sale a las 10 y 3 de la noche y llega a las 7 de la mañana. Desea algo más*”.

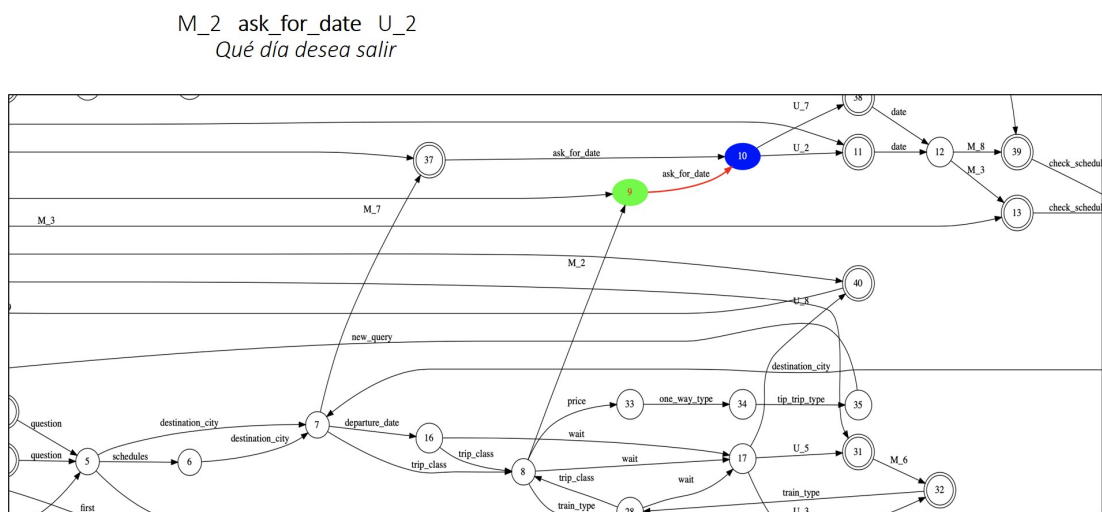


Figura 5.7: Respuesta inferida por el autómata: “*ask\_for\_date*”.

Como puede observarse en estas pruebas, las secuencias semánticas que infiere el autómata a través del modelo de inferencia gramatical es lo que conforma la plantilla de respuesta a las preguntas que se le hacen al sistema, las cuales se utilizan en el módulo de generación de respuesta. Estos resultados se reportaron también en [Vázquez et al. (2020b)].

### 5.3.4. Módulo de generación de respuesta

Finalmente en el módulo para la generación de la respuesta al usuario se utiliza la plantilla de respuesta generada en el módulo de inferencia gramatical, la cual sirve

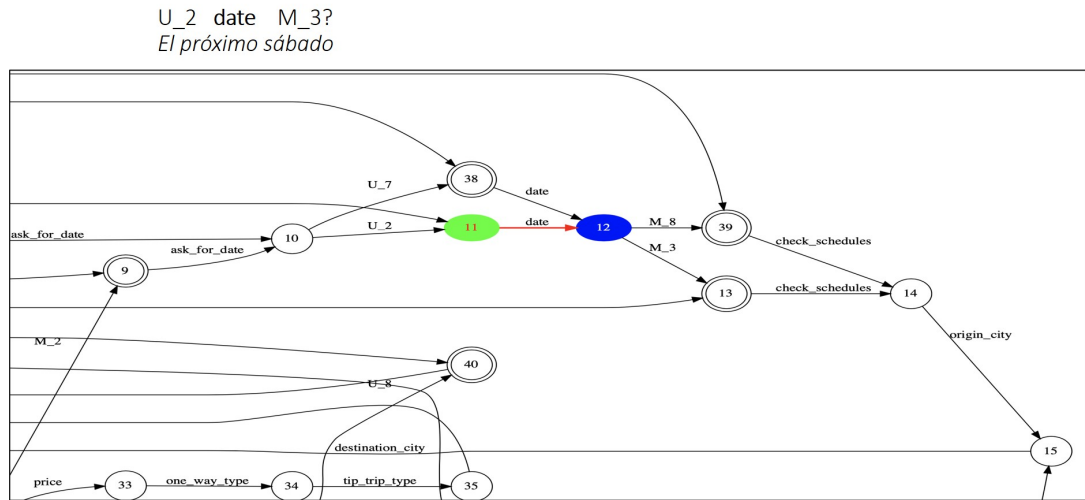


Figura 5.8: Reconocimiento de la etiqueta semántica “date” que corresponde al diálogo “El próximo sábado” .

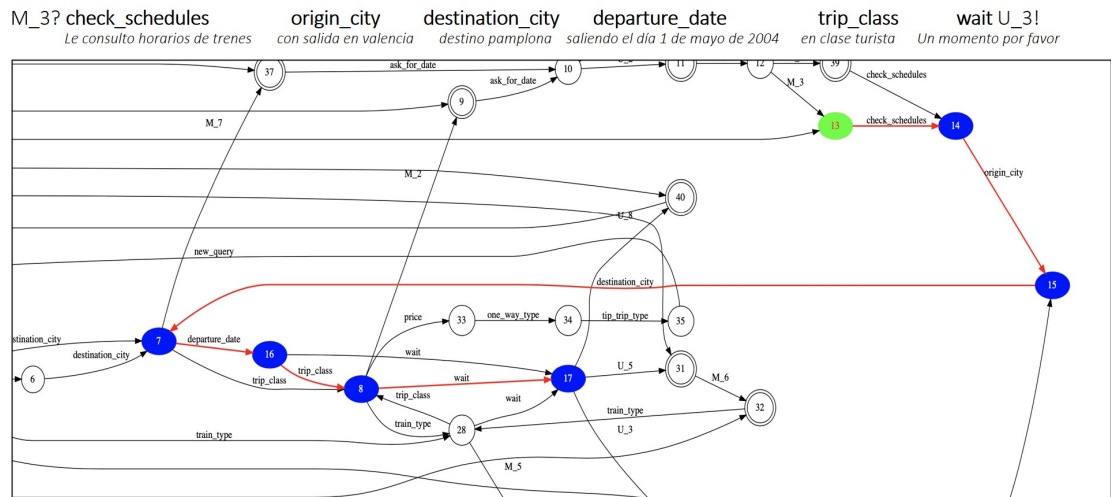


Figura 5.9: Respuesta inferida por el autómata “check\_schedules origin\_city destination\_date trip\_class wait” que corresponde al diálogo: “le consulto horarios de trenes con salida en valencia destino pamplona saliendo el día 1 de mayo de 2004 en clase turista. Un momento por favor” .

como base para la obtención de los valores de respuesta a partir de las consultas a una ontología, la cual fue diseñada explícitamente para el caso de estudio, tal y como se mostró en la Sección 4.4.1.

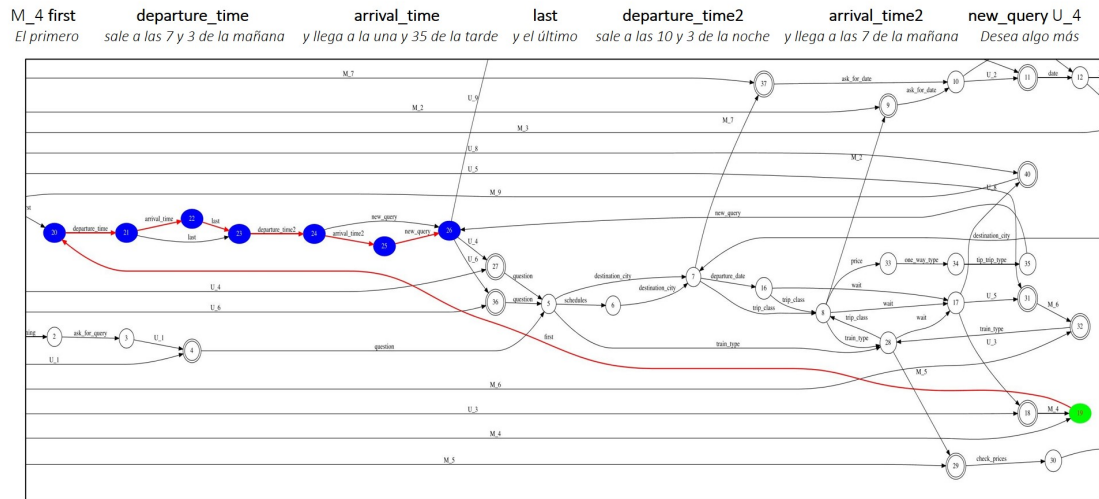


Figura 5.10: Respuesta inferida por el autómata en una segunda intervención del sistema “*first departure\_time arrival\_time last departure\_time2 arrival\_time2 new query*” que corresponde al diálogo: “*El primero sale a las 7 y 3 de la mañana y llega a la una y 35 de la tarde y el lunes sale a las 10 y 3 de la noche y llega a las 7 de la mañana. Desea algo más*”.

### 5.3.4.1. Descripción de la prueba

Para poder realizar las pruebas y siguiendo con los diálogos de prueba se realizaron las consultas a la ontología utilizando como base las plantillas de respuesta obtenidas por el módulo de inferencia gramatical. Estas plantillas de respuesta, como hemos mencionado anteriormente, son una secuencia de entidades semánticas que se utilizan para realizar las consultas en SPARQL en la ontología.

La representación de la base de conocimientos mediante una ontología se implementó utilizando la herramienta Protégé- 5.0.0. La ontología contiene información del mundo real que permiten construir las respuestas finales, mediante el proceso de asociación de los valores obtenidos con las entidades semánticas presentes en la plantilla de respuesta.

En particular, la ontología contiene la información del corpus DIHANA, distribuida en cinco clases: *CIUDAD*, *LINEA*, *RECORRIDO*, *TIPO\_DE\_VIAJE* y *TREN* cada una de estas clases contiene instancias con la información específica tomada de los diálogos del corpus. En la Figuras 5.11 y 5.12 se puede ver de manera general la



estructura que tiene la ontología.

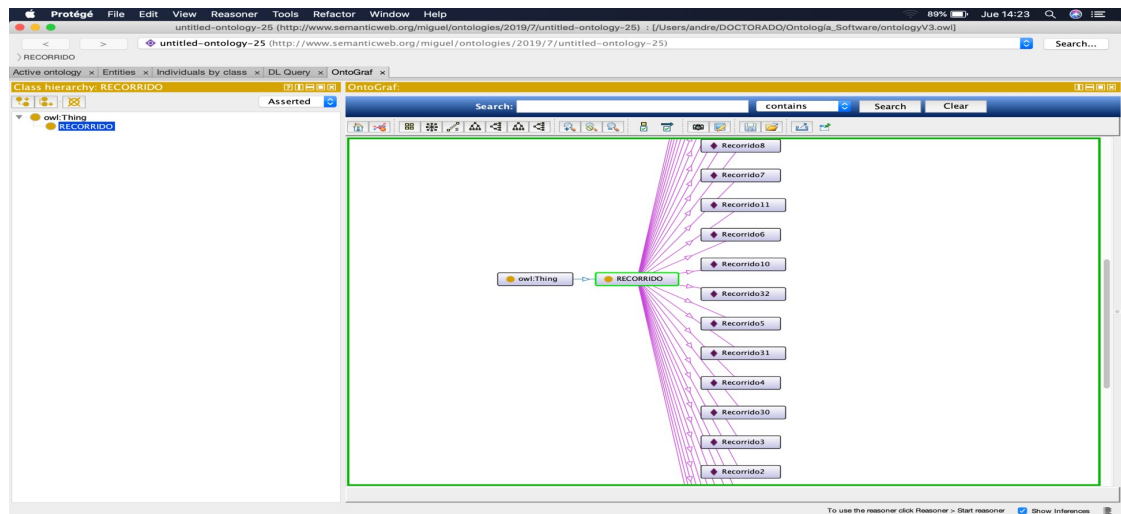


Figura 5.11: Se muestra la clase *RECORRIDO* con sus instancias correspondientes (recorridos).

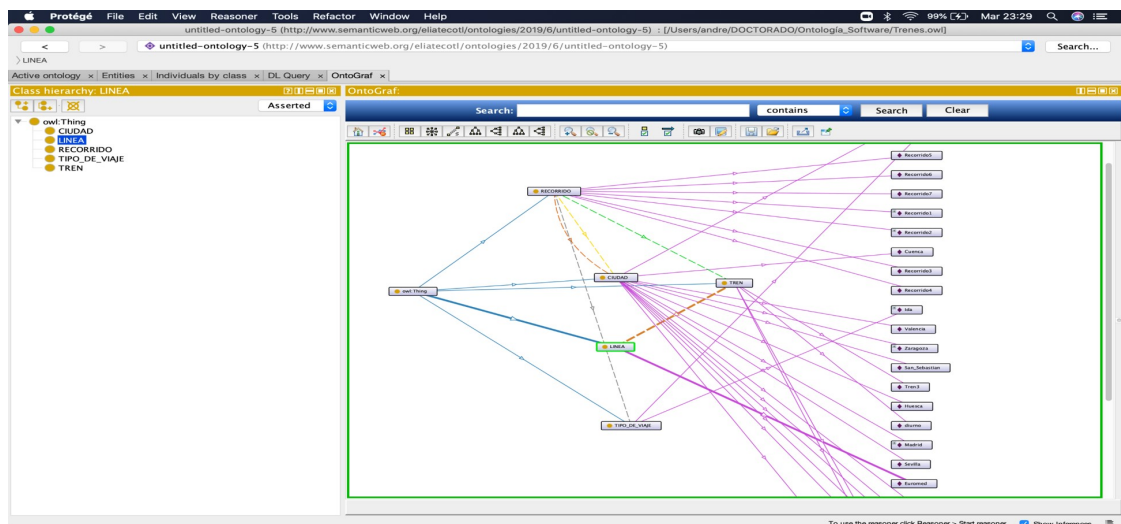


Figura 5.12: Se muestra la ontología con las cinco clases.

Al inicio de esta sección se mencionó que la ontología se diseñó y se implementó de acuerdo a la necesidad que se tenía, es decir poder manejar adecuadamente dentro de la ontología las secuencias de entidades semánticas que forman la plantilla de respuesta y así para poder obtener la información que representa a cada entidad semántica, para obtener esta información se hacen consultas a la ontología utilizando

*Query SPARQL*. En la Figura 5.13 se muestra un ejemplo de una consulta en donde se ve claramente el uso de las etiquetas semánticas, mientras que en la Figura 5.14 se muestra la respuesta a esta consulta.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX sis: <http://www.semanticweb.org/miguel/ontologies/2019/7/untitled-ontology-25#>

SELECT DISTINCT (?f AS ?fecha_salida) (?c AS ?hora_salida) (?p AS ?ciudad_origen) (?o AS
?ciudad_destino)

WHERE {

    ?x sis:tieneFechaRecorrido ?f.
    FILTER (?f = "viernes día 5 de marzo de 2004"^^xsd:string).
    ?x sis:tieneHoraSalida ?c.
    ?x sis:tieneCiudadOrigen ?p.
    FILTER(?p = "Zaragoza"^^xsd:string).
    ?x sis:tieneCiudadDestino ?o.
    FILTER(?o = "Madrid"^^xsd:string)

}

```

Figura 5.13: Ejemplo de consulta a la ontología utilizando SPARQL.

fecha_salida	hora_salida	ciudad_origen	ciudad_destino
'viernes día 5 de marzo de 2004'^^<http://www	"09:00:00"^^<http://www.w3.org/2001/XMLSchema#	"Zaragoza"	"Madrid"
'viernes día 5 de marzo de 2004'^^<http://www	"19:10:00"^^<http://www.w3.org/2001/XMLSchema#	"Zaragoza"	"Madrid"
'viernes día 5 de marzo de 2004'^^<http://www	"02:48:00"^^<http://www.w3.org/2001/XMLSchema#	"Zaragoza"	"Madrid"

Figura 5.14: Ejemplo de Respuesta de la consulta a la ontología, en la parte superior de cada columna de la respuesta se muestra el uso de las entidades semánticas.

#### 5.3.4.2. Evaluación de la prueba

Las respuestas mostradas en lenguaje natural fueron generadas por el módulo de verificación y estructura de respuesta. Los valores de respuesta son obtenidos por consulta a la ontología, tal y como se explicó en la Sección 4.4.1. Usando consultas SPARQL, se obtienen los segmentos de diálogos correspondientes a las secuencias de entidades semánticas, las cuales se estructuran siguiendo el contexto de la pregunta correspondiente para generar la respuesta de manera correcta.

# Capítulo 6

## Conclusiones

En este capítulo se exponen las principales conclusiones del trabajo realizado, indicando si se han alcanzado los objetivos planteados al comienzo de la tesis doctoral. Así mismo, se presentan nuevos planteamientos de investigación derivados directamente del trabajo realizado en esta tesis.

La conclusión fundamental que se puede extraer de esta tesis doctoral es el haber obtenido un *modelo para la generación automática de diálogos y la implementación de este modelo en el sistema de diálogos*.

Otro de los objetivos planteados al inicio de este trabajo fue el de determinar el dominio de trabajo para la generación automática de diálogos, lo cual se logró pues aquí se utilizó el corpus DIHANA cuyo dominio es el de consulta de horarios, precios y tipos de trenes. Este corpus se etiquetó semánticamente con la finalidad de poder procesarse dentro del sistema. Así también se desarrollaron los métodos para el preprocesamiento de los diálogos utilizados en el reconocimiento del lenguaje, el cual era otro objetivo planteado en este trabajo de investigación.

Otro objetivo medular para el desarrollo de este trabajo de tesis y que se logró, fue el de implementar un modelo para inferir la gramática asociada a un conjunto de diálogos correspondientes a un dominio específico.

Finalmente, el último objetivo planteado fue el de diseñar pruebas para la evaluación del modelo computacional para la generación automática de diálogos en un dominio específico. Estas pruebas se realizaron y se llevaron a cabo; de hecho, los

resultados obtenidos fueron reportados en diferentes publicaciones ya mencionadas en este trabajo de tesis.

Cabe mencionar que con la intención de verificar el funcionamiento del sistema, aplicando el modelo propuesto, se realizó una traducción de las etiquetas semánticas al idioma inglés. Esto implicó el tener que generar nuevamente el modelo de reconocimiento del lenguaje entrenado con las nuevas etiquetas del corpus en Inglés. El modelo de inferencia gramatical también se re-entrenó con las nuevas secuencias de entidades semánticas ahora en inglés y así poder trabajar con el nuevo autómata inferido. Lo mismo se hizo con la ontología, ya que las respuestas obtenidas de la ontología deben de coincidir con las etiquetas semánticas como se explicó en la Sección 4.4.1. El tener esta nueva gramática inferida permitió entender el procedimiento para expandir el uso o explotación del modelo en otro idioma.

## 6.1. Contribuciones

En este trabajo de investigación doctoral se han estudiado todos los procesos involucrados en la tarea de conformar un sistema de diálogos para el Español en el dominio específico de preguntas sobre dudas al acceso a un sistema de transporte basado en trenes.

Se han propuesto diversos procedimientos y técnicas para conformar una metodología completa para la gestión del diálogo entre humanos y computadoras. A continuación se resumen las principales contribuciones del trabajo realizado en la presente tesis doctoral:

1. Una **primera contribución** consistió en adaptar el concepto del reconocimiento de entidades nombradas para poder identificar entidades semánticas en los diálogos y a partir de ahí, traducir las oraciones escritas en lenguaje natural a una secuencia de entidades semánticas que pudiesen ser procesadas automáticamente por un sistema computacional. La aproximación propuesta fue aplicada en corpus de dominio específico, resultado en un módulo de reco-

nocimiento del lenguaje que no había sido reportado en la literatura. Derivado de esta propuesta, se publicó un artículo en una revista indizado por Scopus.

2. Una **segunda contribución** de este trabajo de investigación es la de generar un modelo de inferencia gramatical que reconoce secuencias de entidades semánticas. La introducción de un componente de vinculación entre intervenciones del usuario y las intervenciones de la computadora permitió concentrar en una misma gramática inferida las dos tipos de intervenciones. Así, se ejecuta un procedimiento de reconocimiento de cadenas aceptables en un grafo, para inmediatamente proceder a colocar el procedimiento en modo generativo, para proporcionar como salida una cadena de la misma gramática, pero que corresponde a una plantilla de respuesta. Este componente innovador, permite obtener una secuencia de intervenciones coherentes en los diálogos.
3. La **tercera contribución** tiene que ver con el diseño e implementación de la ontología que representa la base de conocimiento de los diálogos. Esta ontología contiene la información del corpus de diálogos DIHANA distribuida en diferentes clases. Es importante resaltar que el diseño e implementación se realizó de acuerdo al uso que iba a tener dentro del sistema, es decir, utilizar una estructura del conocimiento en el que se utilizan las mismas etiquetas semánticas que se encuentran presentes en la plantilla de respuesta, a fin de garantizar un enlace perfecto entre las respuestas obtenidas desde la ontología para con la plantilla de respuesta.

Finalmente, y respondiendo a las preguntas de investigación que se habían planteado al inicio de este trabajo de tesis, se considera lo siguiente:

**PREGUNTA DE INVESTIGACIÓN 1:** ¿En qué medida es posible generar diálogos automáticamente mediante el uso de técnicas basadas en aprendizaje automático?

**RESPUESTA:** SI es posible y factible generar diálogos automáticamente utilizando técnicas basadas en aprendizaje automático como las utilizadas en este

trabajo de tesis. A través de este documento de tesis se ha mostrado como las técnicas basadas en aprendizaje automático han podido ser aplicadas, específicamente en los módulos de reconocimiento del lenguaje y en el de inferencia gramatical.

**PREGUNTA DE INVESTIGACIÓN 2:** ¿El dominio de trabajo influye para la generación automática de diálogos?

**RESPUESTA:** SI, para la propuesta metodológica planteada en este trabajo de tesis, el dominio de trabajo influye de manera sustancial para la generación automática de diálogos. Se requiere en particular de un corpus de diálogos del dominio de trabajo con sus correspondientes etiquetas semánticas para poder llevar a cabo la generación automática de diálogos en ese dominio.

**PREGUNTA DE INVESTIGACIÓN 3:** ¿Qué factores influyen en el método para la obtención de los datos de entrada (diálogos entre personas)?

**RESPUESTA:** El procedimiento para la adquisición de los datos de entrada no pudo ser verificado en este trabajo de tesis, debido principalmente a que se trabajó con un corpus de diálogos preconstruido. Se concluyó que el proceso mismo de construcción de un corpus etiquetado semánticamente requeriría un esfuerzo en tiempo que pondría en riesgo la elaboración de las propuestas para los módulos del sistema de diálogo propuesto.

**PREGUNTA DE INVESTIGACIÓN 4:** ¿De qué manera el etiquetado del corpus impacta en la generación de los diálogos?

**RESPUESTA:** El etiquetado semántico del corpus es fundamental en el proceso de generación de diálogos. Permite definir las entidades semánticas que generalizan la conceptualización de las intervenciones del usuario, facilitando los procedimientos de reconocimiento del lenguaje, inferencia gramatical y generación de la plantilla de respuesta. Sin un etiqueta semántico, no sería posible llevar a cabo el proceso de generación de diálogos bajo la perspectiva del modelo planteado en este trabajo de tesis.

**PREGUNTA DE INVESTIGACIÓN 5:** ¿En qué medida es posible generar pruebas para la evaluación del modelo computacional desarrollado?

**RESPUESTA:** Tal y como se puede observar en el Capítulo de pruebas y resultados, se consideró viable llevar a cabo las evaluaciones de los módulos inmersos en el sistema global desarrollado. En algunos casos a través de medidas de exactitud, y en otros casos a través de medidas de precisión.

En resumen, en este trabajo de tesis doctoral se ha presentado una aproximación completa como un modelo para la generación automática de diálogos, el cual se pueda emplear en otros contextos de diálogos y en otros idiomas. Los procedimientos de adecuación se encuentran principalmente en la obtención de corpora etiquetado para la tarea y el idioma específico. A partir de ahí, se pueden aplicar exactamente los mismos procedimientos para liberar un sistema de diálogos útil en algún otro idioma a dominio de aplicación.

## 6.2. Trabajo futuro

Abordar un sistema completo para la gestión de diálogos ha sido una empresa con muchos retos. Ha sido necesario proponer técnicas diversas para manejar la información desde la intervención de un usuario a fin de darle respuesta a través de un diálogo coherente. Por esta razón existen aun diversas líneas de trabajo que pueden ser estudiadas, desarrolladas y verificadas en el futuro. A continuación, se exponen brevemente algunas ideas que pueden ser consideradas como punto de partida para desarrollar proyectos de investigación asociados con la generación automática de diálogos.

Una de las posibles líneas de investigación podría ser el de atacar el problema de inferencia gramatical utilizando modelos de redes neuronales, los cuales han mostrado un buen desempeño en diferentes aplicaciones en el área del procesamiento del lenguaje natural.

Otro trabajo futuro que podría ser considerado como una mejora a este trabajo de tesis, sería el de poder añadir los módulos de traducción de voz a texto y de texto a voz, tanto en la entrada como en la salida del sistema de diálogo.

Debido a la complejidad de los componentes involucrados en un sistema de diálogos y su aplicación en diversos idiomas y dominios de interés, es muy difícil llevar a cabo un procedimiento de comparación de rendimientos. Consideramos que este tema podría ser investigado a futuro, para generar ambientes de comparación independientes del dominio que puedan permitir determinar el rendimiento de los procesos desarrollados, independientemente de las técnicas empleadas. En este trabajo de tesis doctoral, se propuso una evaluación basada en los componentes, más que en el resultado global. Y este tipo de evaluación permitió solventar de manera parcial el mecanismo de evaluación por comparación con otros trabajos reportados en la literatura. Sin embargo, aún existe mucho trabajo por realizar en este sentido.

Finalmente, la utilización de la metodología planteada en otro idioma o dominio de interés requeriría, tal y como se ha mencionado anteriormente, un esfuerzo importante para la construcción del corpus de entrenamiento, por lo que existe ahí un espacio de desarrollo en el ámbito de la generación de diálogos.



# Bibliografía

- Alpaydin, E. (2010). *Introduction to Machine Learning*. The MIT Press, 2nd edition.
- Barlier, M., Pérolat, J., Laroche, R., and Pietquin, O. (2015). Human-machine dialogue as a stochastic game. In *Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic*, pages 2–11.
- Becerra-Bonache, L. (2008). Aproximación de la teoría de la inferencia gramatical a los estudios de adquisición del lenguaje. pages 327 – 338.
- Benedí, J.-m., Lleida, E., Varona, A., José Castro, M., Galiano, I., Justo, R., Letona, I. L. D., and Miguel, A. (2006). Design and acquisition of a telephone spontaneous speech dialogue corpus in spanish: Dihana. In *In Fifth LREC*, pages 1636–1639.
- Castro, L. A. and Rodríguez, M. (2018). Interacción humano-computadora y aplicaciones en México. *México: Academia Mexicana de Computación*.
- Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124.
- Chomsky, N. (1957). *Syntactic Structures*. Mouton publishers, The Hague, 1st edition.
- Chomsky, N. (1969). *Aspects of the Theory of Syntax*. The MIT Press. MIT Press.
- Dusek, O. and Jurčicek, F. (2016). A context-aware natural language generator for dialogue systems. *CoRR*, abs/1608.07076.

- Fu, K. S. (1982). *Syntactic pattern recognition and applications*. Englewood Cliffs, N.J Prentice-Hall, 1st edition.
- Garcia, P. and Vidal, E. (1990). Inference of k-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):920–925.
- Gold, E. M. (1967). Language identification in the limit. *Information and Control*, 10(5):447 – 474.
- Grishman, R. and Sundheim, B. (1995). Design of the muc-6 evaluation. In *Proceedings of the 6th conference on Message understanding*, pages 1–11. Association for Computational Linguistics.
- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199 – 220.
- Harper, R., Rodden, T., Rogers, Y., and Sellen, A. (2008). *Being Human: Human-Computer Interaction in the Year 2020*. Microsoft Research Ltd, 1st edition.
- Iijima, S. and Kobayashi, I. (2016). A pomdp-based multimodal interaction system using a humanoid robot. In *Proceedings of the 30th Pacific Asia Conference on Language, Information and Computation, PACLIC 30, Seoul, Korea, October 28 - October 30, 2016*, pages 519–525.
- Karray, F., Alemzadeh, M., Abou Saleh, J., and Arab, M. N. (2008). Human-computer interaction: Overview on state of the art. *International Journal on Smart Sensing and Intelligent Systems*, 1:137–159.
- Kleene, S. C. (31 Dec. 1956). *Representation of Events in Nerve Nets and Finite Automata*, pages 3 – 42. Princeton University Press, Princeton.
- Konstantopoulos, S. (2010). An embodied dialogue system with personality and emotions. In *Proceedings of the 2010 Workshop on Companionable Dialogue Systems*, pages 31–36. Association for Computational Linguistics.

- Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, Williams College, Williamstown, MA, USA, June 28 - July 1, 2001, pages 282–289.
- Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., and Jurafsky, D. (2016). Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, November 1-5, 2016*, volume abs/1606.01541, pages 1191–1202, Austin, Texas,. Association for Computational Linguistics.
- Li, X. and Boyer, K. (2016). Reference resolution in situated dialogue with learned semantics. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 329–338.
- Linz, P. (2011). *An Introduction to Formal Languages and Automata*. Jones & Bartlett Learning, 1st edition.
- Lison, P. (2012). Probabilistic dialogue models with prior domain knowledge. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL '12*, pages 179–188, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Lison, P. (2013). Model-based bayesian reinforcement learning for dialogue management. *Proceedings of the International Conference on Spoken Language Processing*, pages 475–479.
- Litman, D. and Forbes-Riley, K. (2014). Evaluating a spoken dialogue system that detects and adapts to user affective states. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 181–185. Association for Computational Linguistics.

- Llisterri, J., Machuca Ayuso, M. J., and Bonafonte, A. (2006). *Los sistemas de diálogo*. Fundación Duques de Soria, Universidad Autónoma de Barcelona. 1a edición.
- Luan, Y., Ji, Y., and Ostendorf, M. (2016). LSTM based conversation models. *CoRR*, abs/1603.09457.
- M. Chantal, P. H. (2002). El concepto de corpus y su definición. <http://elies.rediris.es/elies18/23.html>. Accedido 01-10-2017.
- Mahadik, S. K. (2017). Dialogue manager for spoken dialogue system : Review.
- Miguel, A., Galiano, M., Tormo, R., Hurtado Oliver, L., Sánchez, J.-A., and Arnal, E. (2003). La plataforma de adquisición de diálogos en el proyecto dihana.
- Misu, T., Georgila, K., Leuski, A., and Traum, D. (2012). Reinforcement learning of question-answering dialogue policies for virtual museum guides. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGDIAL '12*, pages 84–93, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1st edition.
- Muñoz, J., Caridad Hernández, Y., Bustos Amador, V., Aranda Chavarría, A., Calderón, M., Collazos, C., Alexandra Méndez, Y., Solano, A., Mendoza, J., Rodríguez, F., Mendoza González, R., Gonzalez Calleros, J., Manuel Gonzales Callero, J., Rodríguez Vizzuett, L., Granollers, T., Gil, R., and Céspedes-Hernández, D. (2014). *Temas de diseño en Interacción Humano-Computadora*.
- Pietquin, O. (2013). Inverse reinforcement learning for interactive systems. In *Proceedings of the 2Nd Workshop on Machine Learning for Interactive Systems: Bridging the Gap Between Perception, Action and Communication, MLIS '13*, pages 71–75, New York, NY, USA. ACM.

- Pinto, D., McCallum, A., Wei, X., and Croft, W. B. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR '03*, pages 235–242, New York, NY, USA. ACM.
- Png, S. and Pineau, J. (2011). Bayesian reinforcement learning for pomdp-based dialogue systems. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*, pages 2156–2159.
- Sadick Ngula, R. (2014). Corpus linguistics and language development in ghana. *Asian Journal of Humanities and Social Studies*, 2(4):522–534.
- Segarra, E., Sanchis, E., Galiano, M., García, F., and Hurtado, L. F. (2002). Extracting semantic information through automatic learning techniques. *IJPRAI*, 16(3):301–307.
- Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J., Gao, J., and Dolan, B. (2015). A neural network approach to context-sensitive generation of conversational responses. *CoRR*, abs/1506.06714.
- Su, P., Gasic, M., Mrksic, N., Rojas-Barahona, L. M., Ultes, S., Vandyke, D., Wen, T., and Young, S. J. (2016). Continuously learning neural dialogue management. *CoRR*, abs/1606.02689.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.
- The Stanford Natural Language Processing Group, . (2011). Stanford named entity recognizer (ner). <https://nlp.stanford.edu/software/CRF-NER.html>. Accedido 10-06-2018.
- Vázquez, A., Pinto, D., Lavalle, J., Jiménez-Salazar, H., and Vilariño, D. (2020a). Grammatical inference of semantic components in dialogues. *Computación y Sistemas*, 24(2).

- Vázquez, A., Pinto, D., Pallares, J., de la Rosa, R., and Tecotl, E. (2020b). Automatic generation of dialogues based on grammatical inference and the use of a knowledge base. *J. Intell. Fuzzy Syst.*, 39(2):2105–2113.
- Vinyals, O. and Le, Q. V. (2015). A neural conversational model. *CoRR*, abs/1506.05869.
- Vázquez, A., Pinto, D., and Vilariño, D. (2018). Identificación de etiquetas semánticas para su uso en diálogos. *Procesamiento del Lenguaje Natural*, pages 99 – 107.
- Wen, T., Gasic, M., Mrksic, N., Su, P., Vandyke, D., and Young, S. J. (2015). Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *CoRR*, abs/1508.01745.
- Yao, K., Zweig, G., and Peng, B. (2015). Attention with intention for a neural network conversation model. *CoRR*, abs/1510.08565.
- Young, S., Gašić, M., Thomson, B., and Williams, J. D. (2013). Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.
- Yuan, X. E. (2014). Named entity recognition using crf++ and c++. <https://github.com/xingdi-eric-yuan/nerpp>. Accedido 15-06-2018.