

Universidad Politécnica de Valencia

Departamento de Sistemas Informáticos y Computación



**UNIVERSIDAD
POLITECNICA
DE VALENCIA**

Sistemas de diálogo basados en modelos estocásticos

Tesis Doctoral

Autor:

Francisco Torres Goterris

Directores:

Dr. D. Emilio Sanchis Arnal

Dra. Dña. Encarna Segarra Soriano

Valencia, Mayo de 2006

Universidad Politécnica de Valencia

Departamento de Sistemas Informáticos y Computación



**UNIVERSIDAD
POLITECNICA
DE VALENCIA**

Sistemas de diálogo basados en modelos estocásticos

D. Francisco Torres Goterris

**Memoria para optar al grado de Doctor en Informática
bajo la dirección de los doctores**

**Dr. D. Emilio Sanchis Arnal
Dra. Dña. Encarna Segarra Soriano**

Valencia, Mayo de 2006

A mis padres,

porque sólo por ellos
he encontrado motivos
para acabar esta tesis

Agradecimientos

El trabajo que se refleja en las líneas de esta tesis se prolonga en un tiempo ya casi indefinido. Y es arriesgado dar nombres en una lista de agradecimientos pues, sin duda, la memoria del que ahora escribe podría sucumbir en lagunas que, más tarde, lamentaría. Eludiré el peligro de los nombres.

Quisiera, desde luego, agradecer el apoyo y colaboración de quienes han estado más cerca en la larga elaboración de la presente memoria. Ahí figuran los directores de la tesis; ahí figuran, sin duda en menor medida, muchos otros. Ya sean compañeros de grupo de investigación, de departamento, de universidad; ya sean de otros grupos, otros departamentos, otras universidades. Ya sean catedráticos, ya sean becarios. Detallar sus respectivos méritos es una tarea acaso titánica, acaso quimérica, y, en esta mañana en la que escribo, me siento algo agotado.

Quisiera agradecer la amistad de aquellos a los que envié la serie de mensajes titulada *El largo y tortuoso camino del doctorando*. No es preciso que les nombre. Ellos saben quienes son. Y yo me sé de memoria la lista. La lealtad auténtica no necesita pregoneros. Quisiera agradecer, igualmente, el cariño o la amistad (según los casos) de los que también leyeron (o les leí) mis mensajes. Quisiera agradecer y recordar, también, a los que no pudieron leerlos, y que tanto hubiera querido.

Quisiera agradecer, muy especialmente, a los miembros de cierta banda, a quienes envié otras muchas series de mensajes, por citar, las tituladas *El sueño del Físico Loco*, *Singulares los tiempos de la lírica*, *El éter*, *La sustancia*, *Casablanca*, *Fragmentaria*, *Dada...* Aunque, tal vez, algunas de sus respuestas llegaran a contrariarme, siempre, en el fondo, me alegré de sus envíos. Y aunque siga pensando que jamás formaré parte de un club que admita como socio a alguien como yo, si tuviera que hacer una excepción, no lo dudaría ni un instante.

Resumen

En la presente tesis, titulada “Sistemas de diálogo basados en modelos estocásticos”, se expone el estado del arte en el área de los sistemas de diálogo y se presenta el trabajo realizado en el diseño e implementación de los módulos de un sistema de diálogo determinado.

La tesis se centra en el estudio de la gestión de diálogo desde una aproximación estadística. La tesis aporta el desarrollo de un sistema de diálogo completo (con entrada y salida de texto, en lengua española, y para una tarea de dominio semántico restringido, la definida en el proyecto de investigación BASURDE). Dicho sistema está constituido por los módulos de comprensión del lenguaje natural, de gestión del diálogo y de generación de respuestas en lenguaje natural. Dado el objetivo central de la tesis, el desarrollo del módulo gestor de diálogo ha sido el principal trabajo y, en consecuencia, es expuesto con la máxima amplitud en la presente memoria.

El limitado tamaño del corpus de diálogos de la tarea BASURDE ha supuesto una severa dificultad en el desarrollo de un gestor de diálogo basado exclusivamente en modelos estadísticos. El módulo gestor de diálogo finalmente implementado determina su estrategia de diálogo mediante la combinación de varias fuentes de conocimiento: unas de carácter estocástico, los modelos aprendidos a partir del corpus; otras de carácter heurístico, reglas que incorporan conocimiento pragmático y semántico, ya sea genérico o específico de la tarea.

Por último, se ha considerado la simulación de los usuarios como una técnica alternativa para fines como la evaluación del comportamiento del sistema de diálogo, la ampliación del corpus mediante diálogos sintéticos, o el aprendizaje dinámico de los modelos estocásticos de diálogo. Se han diseñado e implementado los correspondientes módulos simuladores de usuario, estudiándose las posibilidades de esta técnica.

Resum

En la present tesi, titulada “Sistemes de diàleg basats en models estocàstics”, s'exposa l'estat de l'art en l'àrea dels sistemes de diàleg i es presenta el treball realitzat en el disseny i implementació dels mòduls d'un sistema de diàleg determinat.

La tesi se centra en l'estudi de la gestió de diàleg des d'una aproximació estadística. La tesi aporta el desenvolupament d'un sistema de diàleg complet (amb entrada i sortida de text, en llengua espanyola, i per a una tasca de domini semàntic restringit, la definida en el projecte d'investigació BASURDE). El dit sistema està constituït pels mòduls de comprensió del llenguatge natural, de gestió del diàleg i de generació de respostes en llenguatge natural. Donat l'objectiu central de la tesi, el desenvolupament del mòdul gestor de diàleg ha sigut el principal treball i, en conseqüència, és exposat amb la màxima amplitud en la present memòria.

La limitada grandària del corpus de diàlegs de la tasca BASURDE ha suposat una severa dificultat en el desenvolupament d'un gestor de diàleg basat exclusivament en models estadístics. El mòdul gestor de diàleg finalment implementat determina la seua estratègia de diàleg per mitjà de la combinació de diverses fonts de coneixement: unes de caràcter estocàstic, els models apresos a partir del corpus; altres de caràcter heurístic, regles que incorporen coneixement pragmàtic i semàntic, ja siga genèric o específic de la tasca.

Finalment, s'ha considerat la simulació dels usuaris com una tècnica alternativa per a finalitats com l'avaluació del comportament del sistema de diàleg, l'ampliació del corpus per mitjà de diàlegs sintètics, o l'aprenentatge dinàmic dels models estocàstics de diàleg. S'han dissenyat e implementat els corresponents mòduls simuladors d'usuari, estudiant-se les possibilitats d'aquesta tècnica.

Abstract

In the present thesis, entitled “Dialog systems based on stochastic models”, the state of the art in the field of the dialog systems is exposed and the work done in the design and implementation of the modules of a specific dialog system is presented.

The thesis is focused on the study of the dialog management following a statistical approach. The development of a complete dialog system (with textual input and output, in Spanish language, and for a task in a restricted semantic domain, the task defined in the BASURDE research project) is the main contribution of the thesis. Such a system is composed by the natural language understanding module, the dialog manager module and the natural language reply generator module. Given the central objective of the thesis, the development of the dialog manager module has been the main work and, in consequence, it is exposed with the maximum amplitude in the present thesis.

The limited size of the dialog corpus of the BASURDE task has supposed a severe difficulty in the development of a dialog manager based exclusively on stochastic models. The dialog manager module that is been finally implemented determines its dialog strategy by means of a combination of several knowledge sources: some of them are of stochastic type, as the models learnt from the corpus; others are of heuristic type, as the rules that incorporate pragmatic and semantic knowledge, either generic or specific to the task.

Finally, the user simulation has been considered as an alternative technique to deal with the evaluation of the behaviour of the dialog system, the extension of the corpus by means of synthetic dialogs, and the dynamic learning of the stochastic dialog models. The corresponding user simulator modules has been designed and implemented, studying the possibilities of this technique.

Índice general

1. Introducción.....	19
2. Estado del arte	25
2.1. Sistemas de diálogo	27
2.1.1. Aplicación de métodos basados en reglas	28
2.1.2. Aplicación de métodos estocásticos.....	30
2.2. Comprensión del lenguaje.....	34
2.2.1. Uso de medidas de confianza.....	42
2.3. Gestor del diálogo.....	47
2.4. Generación de respuestas.....	60
2.5. Evaluación de los sistemas de diálogo	63
2.6. Simulación de diálogos.....	68
2.6.1. Adaptación a usuario y/o dominio	73
3. Descripción de la tarea BASURDE.....	77
3.1. Introducción	79
3.2. Adquisición del corpus	79
3.3. Etiquetado del corpus: representación semántica mediante frames.....	81
3.4. Etiquetado del corpus: actos de diálogo.....	82
4. Módulo de comprensión	85
4.1. Introducción.....	87
4.2. Descripción del módulo de comprensión.....	88
4.3. Incorporación de otras fuentes de información	92
4.4. Evaluación	93
4.5. Comparativa y conclusiones	94
5. Gestor de diálogo.....	97
5.1. Introducción	99
5.2. Entrada al gestor de diálogo.....	99
5.3. Aprendizaje de los modelos de diálogo.....	101
5.4. Algoritmo del gestor de diálogo.....	105
5.5. El registro histórico.....	106
5.6. Generalización semántica y transiciones a estados de usuario.....	107
5.7. Transiciones a estados de sistema.....	113
5.8. Medidas de confianza y transiciones en el modelo	115
5.9. Ejemplo de diálogo.....	118
6. Generador de respuestas	127
6.1. Descripción de los frames del sistema	129
6.2. Descripción del generador de respuestas	131
6.3. Confirmación implícita y medidas de confianza	135
7. Simulación de usuarios.....	139
7.1. Introducción	141
7.2. Arquitectura del sistema y algoritmo del simulador.....	141
7.3. Reglas de selección de transiciones de usuario	145
7.4. Generación de los frames de usuario	146
7.5. Modificaciones en el gestor de diálogo.....	148
7.6. Ejemplo de diálogo	149
7.7. Generador de respuestas del simulador de usuario.....	158

8.	Evaluación del sistema de diálogo.....	161
8.1.	Introducción	163
8.2.	Evaluación del gestor de diálogo	164
8.3.	Ejemplos del comportamiento del sistema.....	164
8.4.	Evaluación experimental.....	167
8.5.	Evaluación / entrenamiento de los modelos (con simulación de usuario).....	170
8.6.	Evaluación de la generalización semántica y la poda de transiciones	184
8.7.	Apunte sobre evaluación del gestor de diálogo y los modelos MD y MD*	186
9.	Conclusiones	191
10.	Anexos.....	199
10.1.	Generador de respuestas	201
10.2.	Simulador de usuario	208
10.3.	Generador de respuestas del simulador de usuario	225
11.	Referencias	233

Índice de figuras

3.1: Ejemplo de representación semántica	82
3.2: Ejemplo de etiquetado de actos de diálogo	84
4.1: Etiquetado de comprensión	88
4.2: Ejemplo de etiquetado de actos de diálogo	88
4.3: Algoritmo del módulo de comprensión	90
5.1: Interconexión del gestor de diálogo	100
5.2: Ejemplo de representación semántica	100
5.3: Ejemplo de etiquetado de actos de diálogo	101
5.4: Algoritmo del gestor de diálogo	105
5.5: Contenido inicial del registro histórico	107
5.6: Ejemplo de generalización semántica	112
5.7: Fragmento de diálogo sin uso de medidas de confianza	116
5.8: Fragmento de diálogo usando medidas de confianza	116
5.9: Fragmento de diálogo sin uso de medidas de confianza	117
5.10: Fragmento de diálogo usando medidas de confianza	117
5.11: Fragmento de diálogo usando medidas de confianza	117
5.12: Ejemplo de diálogo, 1º turno	121
5.13: Ejemplo de diálogo, 2º turno	122
5.14: Ejemplo de diálogo, 3º turno	123
5.15: Ejemplo de diálogo, 4º turno	124
5.16: Ejemplo de diálogo, 5º turno	125
6.1: Algoritmo del generador de respuestas	132
6.2: Fragmento de diálogo	136
7.1: Diagrama de bloques del sistema de diálogo con simulador de usuario	143
7.2: Algoritmo del simulador de usuario	144
7.3: Algoritmo del gestor de diálogo	145
7.4: Reglas aplicadas para elegir estado de usuario en el simulador	146
7.5: Algoritmo de evaluación del éxito de las simulaciones	148
7.6: Ejemplo de diálogo, 1º turno del simulador de usuario	149
7.7: Ejemplo de diálogo, 1º turno del sistema	150
7.8: Ejemplo de diálogo, 2º turno del simulador de usuario	151
7.9: Ejemplo de diálogo, 2º turno del sistema	152
7.10: Ejemplo de diálogo, 3º turno del simulador de usuario	153
7.11: Ejemplo de diálogo, 3º turno del sistema	155
7.12: Ejemplo de diálogo, 4º turno del simulador de usuario	156
7.13: Ejemplo de diálogo, 4º turno del sistema	157
7.14: Ejemplo de diálogo, 5º turno del simulador de usuario	156
8.1: Evolución de las probabilidades de algunas transiciones durante el entrenamiento mediante simulación de diálogos (aprendizaje del modelo MD*)	171
8.2: Ejemplo de diálogo	174
8.3: Evolución de las probabilidades de algunas transiciones durante el entrenamiento mediante simulación con introducción de errores (aprendizaje del modelo MD* ^c)	176
8.4: Evolución de las probabilidades de algunas transiciones durante el entrenamiento mediante simulación con introducción de errores (aprendizaje del modelo MD* ^j)	179
10.1: Evolución de las probabilidades de algunas transiciones durante el entrenamiento de los modelos MD*, MD* ^e y MD* ^j	209

Índice de tablas

3.1. Estadística del corpus BASURDE	80
4.1: Frames y actos de diálogo	89
4.2. Resultados de la evaluación	93
4.3. Efectos de las reglas	94
4.4. Resultados de Hurtado et al., 2004	94
5.1. Estadísticas de modelos de diálogo en función del nivel de etiquetado	103
5.2. Modelos de diálogo antes y después del agrupamiento de estados	104
6.1. Frames del sistema	129
7.1. Frames del usuario	158
8.1. Tipo y cantidad de errores introducidos en las series	167
8.2. Evaluación con WA de 80%	168
8.3. Evaluación con WA de 70%	168
8.4. Evaluación con entrada hablada	169
8.5: Aprendizaje del modelo MD*	170
8.6: Primera comparativa, MD y MD*	172
8.7: Segunda comparativa, MD y MD*	173
8.8: Aprendizaje del modelo MD* ^c	175
8.9: Tercera comparativa, MD, MD* y MD* ^c	177
8.10: Cuarta comparativa, MD y MD* ^c	178
8.11: Aprendizaje del modelo MD* ^j	179
8.12: Comparativa, MD, MD* ^c y MD* ^j	180
8.13: Comparativa, MD, MD* ^c y MD* ^j	181
8.14: Quinta comparativa, MD y MD* ^c	182
8.15: Comparativa sobre generalización semántica	184
8.16: Comparativa sobre poda de transiciones según reglas de consistencia con el RH	185
8.17: Evaluación con usuario real, comparativa de MD y MD* ^c	186
8.17a: Evaluación con usuario real del modelo MD	188
8.17b: Evaluación con usuario real del modelo MD* ^c	189
8.18: Generalización semántica y modelo MD	187
8.19: Generalización semántica y modelo MD* ^c	187
10.1: Frames de sistema generados (listado completo)	201
10.2: Plantillas de generación de respuestas del sistema (listado completo)	206
10.3. Relación de escenarios usados en la simulación de diálogos	208
10.4a: Evaluación del modelo MD* ^c	216
10.4b: Evaluación del modelo MD* ^c	216
10.4c: Evaluación del modelo MD* ^c	217
10.4d: Evaluación del modelo MD* ^c	217
10.5a: Evaluación del modelo MD*	218
10.5b: Evaluación del modelo MD*	218
10.5c: Evaluación del modelo MD*	219
10.5d: Evaluación del modelo MD*	219
10.6a: Evaluación del modelo MD	220
10.6b: Evaluación del modelo MD	220
10.6c: Evaluación del modelo MD	221
10.6d: Evaluación del modelo MD	221
10.7a: Evaluación del modelo MD* ^c	222
10.7b: Evaluación del modelo MD* ^c	222
10.8a: Evaluación del modelo MD	223
10.8b: Evaluación del modelo MD	223
10.9: Evaluación del modelo MD	224
10.10: Evaluación del modelo MD* ^c	224
10.11: Frames de usuario, generados por el simulador de usuario (listado completo)	225
10.12: Plantillas de generación de respuestas del simulador de usuario (listado completo)	230

1. Introducción

1. Introducción

La presente tesis, titulada “Sistemas de diálogo basados en modelos estocásticos”, tiene como objetivos:

- ❑ Exponer el estado actual del desarrollo de los sistemas de diálogo. En especial, los trabajos relativos a la comprensión del lenguaje y la gestión del diálogo llevados a cabo por diferentes grupos de investigación en los últimos años.
- ❑ Exponer el trabajo realizado en el diseño e implementación de los módulos de un sistema de diálogo determinado (en concreto, los módulos de comprensión del lenguaje, de generación de respuestas, y, especialmente, de gestión del diálogo), así como su evaluación tanto por parte de usuarios reales como mediante el uso de módulos simuladores de usuario (módulos que también han sido objeto de diseño e implementación).

Así pues, en esta breve introducción, se precisará qué se entiende por sistemas de diálogo y qué componentes o módulos suelen constituir tales sistemas. En general, un sistema de diálogo sería toda aplicación informática que permita a los usuarios de una computadora comunicarse con dicha computadora de una manera interactiva. Se suele concretar el concepto de sistema de diálogo a las aplicaciones cuyo objetivo es el acceso a información de características bien definidas (información contenida en bases de datos) y cuyo medio de acceso sea vocal, es decir, mediante el habla.

En este campo de las Tecnologías del Habla, un tema de especial interés, y de especial dificultad, es el desarrollo de sistemas de diálogo entre las computadoras y sus usuarios humanos. Conforme la interacción con las computadoras se extiende a todas las áreas de la actividad humana y conforme el acceso a todo tipo de información se automatiza mediante tales máquinas, se incrementa el interés por disponer de sistemas que faciliten esta comunicación persona-máquina del modo más natural posible.

El modo de comunicación entre seres humanos más natural, de uso más extendido, y en vigor desde que existen las lenguas, es la conversación. Idealmente, desearíamos poder interactuar con las computadoras con facilidad, flexibilidad y fluidez pareja a la que caracteriza nuestro diálogo con otras personas. Aunque es probable que nunca se alcancen semejantes prestaciones, los constantes avances de la investigación en Tecnologías del Habla han permitido que sean factibles sistemas de comunicación persona-máquina mediante voz, con cierto grado de flexibilidad (iniciativa mixta en el desarrollo del diálogo) y orientados a la consulta de información muy específica (tareas de dominio semántico restringido).

En cualquier caso, tales sistemas de diálogo deben abordar, en mayor o menor grado, la modelización del conocimiento lingüístico y la simulación del comportamiento lingüístico, y, en consecuencia, se encuentran serias dificultades en su desarrollo. Es habitual un desarrollo modular de los sistemas de diálogo hablado persona-máquina, lo que permite desglosar las dificultades entre los diferentes componentes del sistema. Desde esta perspectiva, los módulos esenciales en todo sistema de diálogo son los siguientes:

- ❑ Módulo de reconocimiento del habla: componente que procesa la voz del interlocutor humano e identifica la secuencia de texto que corresponde a la señal acústica recibida.
- ❑ Módulo de comprensión del lenguaje: componente que recibe la secuencia de texto identificada por el anterior módulo e identifica la información relevante que suministra el texto en el ámbito de la tarea considerada (determina el significado de la intervención del usuario humano).
- ❑ Módulo de gestión del diálogo: componente que recibe el significado de la intervención del usuario (en términos de un lenguaje conceptual o de representación semántica), procedente del anterior módulo, y determina la siguiente intervención del sistema (también mediante una representación semántica) conforme a algún modelo que establezca la estrategia o conducta del sistema ante las diferentes situaciones que se planteen en la tarea considerada.
- ❑ Módulo de generación de respuestas: componente que, a partir de la representación semántica de la intervención del sistema, procedente del anterior módulo, genera una secuencia de texto, en el idioma del usuario, equivalente en su significado.
- ❑ Módulo de síntesis del habla: componente que recibe la respuesta del sistema como texto en lenguaje natural y genera la correspondiente señal de audio, que será la respuesta que llegará al usuario humano, interlocutor del sistema.

En todo sistema de diálogo, una vez determinada cuál será la siguiente intervención del sistema en términos de un lenguaje conceptual, realizar su conversión a texto en lenguaje natural (generación de respuesta) y de éste a voz (síntesis del habla) es un problema, comparativamente, menor. Los principales retos se encuentran en el desarrollo de los otros módulos.

A lo largo de las últimas décadas, en el campo del reconocimiento automático del habla, se ha progresado desde el reconocimiento de palabras aisladas dentro de conjuntos reducidos de vocabulario hasta el reconocimiento de habla continua y con conjuntos de vocabulario cada vez mayores. Los avances también son significativos en cuanto a la correcta identificación de los fonemas en entornos ruidosos. Actualmente, los métodos estocásticos (como los modelos ocultos de Markov) proporcionan las mejores tasas de reconocimiento en estos entornos de acústica deficiente, como es el caso de la locución telefónica, en los que suelen encontrarse los usuarios.

Los problemas de la comprensión del lenguaje natural y del control del diálogo por parte del sistema presentan mayores dificultades. Por ello, la aplicación de los métodos estocásticos ha comenzado a dar resultados en tiempos mucho más recientes y constituye actualmente una vía de investigación innovadora. Anteriormente estos problemas se abordaban con una modelización del conocimiento lingüístico, formalizado mediante reglas deducidas por usuarios expertos en la tarea considerada. Esta vía de investigación, más tradicional, sigue activa.

Los sistemas que se desarrollan según el paradigma deductivo, usando métodos basados en reglas, tienen los inconvenientes del elevado coste del diseño del modelo (dada la necesidad de que participe personal especializado) y de la limitación en su aplicación, ya que el cambio en la tarea del sistema implica el diseño de un nuevo modelo. Los sistemas que se des-

arrollan según el paradigma inductivo, usando métodos estocásticos, presentan la dificultad previa de la adquisición de un corpus. Éste debe proporcionar un número de muestras suficientemente grande como para que sea posible, aplicando técnicas de aprendizaje automático, inferir un modelo que describa adecuadamente la tarea considerada.

Actualmente, tanto los métodos estocásticos como los basados en reglas permiten desarrollar sistemas capaces de comprender automáticamente las intervenciones de las personas, expresadas en su lenguaje natural (siempre y cuando usen palabras del vocabulario considerado en los modelos, palabras pertenecientes al dominio semántico de la tarea modelizada). En cambio, el problema del control o gestión del diálogo por parte del sistema aún no ha sido satisfactoriamente resuelto aplicando métodos estocásticos, debido a la ausencia de corpus de talla suficiente y a la mayor dimensión del espacio de sucesos objeto de modelización. Los sistemas que, hoy en día, gestionan automáticamente los diálogos persona-máquina se basan en la aplicación de reglas o bien son sistemas híbridos que integran ambos tipos de métodos.

En esta tesis, se presentan módulos de comprensión del lenguaje natural y de gestión del diálogo que utilizan modelos estocásticos, aprendidos automáticamente a partir de un corpus de diálogos de la tarea considerada, y también ciertos conocimientos heurísticos sobre la tarea, recopilados en una serie de reglas. En consecuencia, el sistema de diálogo desarrollado puede etiquetarse como un sistema híbrido.

La tesis se estructura del modo siguiente. En la Sección 2 se aborda el estado del arte en lo relativo a los sistemas de diálogo, incidiendo sobre todo en las soluciones dadas a los problemas de la comprensión del lenguaje natural y de gestión del diálogo, y también en los aspectos relativos a la evaluación de los sistemas y al uso de técnicas de simulación de diálogos como medio para evaluar y mejorar los sistemas.

En la Sección 3, se describe la tarea BASURDE, tarea definida en el homónimo proyecto de investigación. En el proyecto BASURDE (1998-2001) participaron las siguientes Universidades: UPC (Politécnica de Cataluña), UPV (Politécnica de Valencia), UJI (Jaume I, Castellón), EHU (País Vasco) y UZ (Zaragoza). El objetivo del proyecto BASURDE era el desarrollo de un sistema de diálogo hablado, en español, para una tarea de dominio semántico restringido: la consulta a una base de datos con información relativa a horarios, precios y servicios en trenes españoles de largo recorrido. Como resultado de dicho proyecto, se adquirió un corpus de diálogo a partir del cual se han aprendido los modelos de comprensión y de diálogo utilizados por los correspondientes módulos del sistema que se presentan en esta tesis. En consecuencia, en esta Sección 3, se describe el etiquetado del corpus de BASURDE.

En la Sección 4, se presenta el módulo de comprensión del lenguaje. Se describe cómo se han integrado en dicho módulo diferentes fuentes de conocimiento: modelos estocásticos, para identificar los conceptos de la tarea, y reglas heurísticas, sobre todo para identificar los atributos de la tarea. Se facilitan los resultados de su evaluación.

En la Sección 5, se presenta el módulo gestor de diálogo. Este módulo es el componente más decisivo para establecer el comportamiento del sistema de diálogo y, también, el componente cuyo desarrollo desde una aproximación estocástica ha resultado más complicado hasta la fecha. Constituye, por tanto, el apartado más relevante de esta tesis. En esta sección 5, se explica cómo se ha llevado a cabo el aprendizaje de los modelos estocásticos de diálogo y cómo se han integrado en la aplicación gestor de diálogo para realizar las funciones

propias de dicho módulo. La estrategia implementada en este gestor de diálogo está determinada tanto por los modelos estocásticos de diálogo como por un conjunto de reglas, que garantizan la consistencia de las respuestas del sistema respecto a la información almacenada en una memoria del gestor de diálogo, que llamamos registro histórico. Se exponen también los procedimientos implementados para soslayar las deficiencias existentes en los modelos estocásticos de diálogo (derivadas de la escasez de muestras de aprendizaje en el corpus), así como la incorporación de las medidas de confianza facilitadas por los módulos previos y su influencia en la respuesta del sistema.

En la Sección 6, se describe el módulo generador de respuestas que se ha desarrollado para interactuar con el gestor de diálogo expuesto en la anterior sección. Se trata de un generador de respuestas basado en el uso de plantillas. En esta sección, además, se expone el formato de los llamados *frames* del sistema (formato de intercambio de información entre el gestor de diálogo y el generador de respuestas) y el uso de las medidas de confianza para controlar la presencia o ausencia de confirmaciones implícitas en las respuestas dadas al usuario.

En la Sección 7, se presenta el módulo simulador de usuario. Este módulo es, en esencia, un gestor de diálogo, pero encargado de generar las intervenciones de los usuarios. Este simulador de usuario tiene un comportamiento estocástico en las transiciones a estados del sistema (cuando ha de procesar la intervención previa del gestor de diálogo del sistema) y un comportamiento determinado por reglas en las transiciones a estados del usuario (cuando ha de generar su propia intervención). Se ha implementado, mediante reglas, una estrategia que simula un usuario cooperativo con el sistema de diálogo. En esta sección, se explica también la arquitectura del sistema de diálogo, extendida para operar con el simulador de usuario, y las modificaciones que fueron precisas en el módulo gestor de diálogo del sistema.

En la Sección 8, se informa de los resultados de la evaluación del sistema de diálogo. El sistema ha sido evaluado tanto con usuarios humanos como con el citado módulo simulador. La sección incluye asimismo los experimentos llevados a cabo para reentrenar, mediante simulación de diálogos, el modelo original (modelo estocástico de diálogo aprendido a partir del corpus BASURDE), en diferentes condiciones (ausencia y presencia de errores en los diálogos), y los experimentos para evaluar los modelos resultantes de tales entrenamientos mediante simulación.

En la Sección 9, se presentan las conclusiones de esta tesis. En la Sección 10, aparecen los anexos, donde se recopila cierta información detallada (tablas, gráficas, etc.) a la que se hace referencia en secciones anteriores. Por último, en la Sección 11, se recopila todas las referencias bibliográficas que se citan a lo largo de la tesis.

2. Estado del arte

2. Estado del arte

2.1 Sistemas de diálogo

En la literatura más reciente pueden encontrarse numerosas referencias sobre sistemas de diálogo hablado (SDS: *Spoken Dialog Systems*) desarrollados por diferentes grupos de investigación (Aust et al., 1995), (Baggia et al., 2000), (Bonafonte et al., 2000), (Córdoba et al., 2001), (CMU, 1999), (Lamel et al., 2000a), (Pieraccini et al., 1997), (Rudnicky et al., 1999b), (Ward y Pellom, 1999), (Zue et al., 2000). Todos estos sistemas coinciden en posibilitar el diálogo con las computadoras en tareas muy definidas, como son las consultas a sistemas automáticos de información. Las tareas más habituales son las relativas a la planificación de viajes: consultas sobre horarios, precios, disponibilidades, etc., en aviones, hoteles, trenes...

La tarea ATIS (*Air Travel Information Services*), de consulta sobre vuelos, sirvió de marco para el desarrollo de los primeros proyectos, como los llevados a cabo en AT&T, concretamente el proyecto AMICA (Pieraccini et al., 1997), donde se aplicaron los métodos estocásticos en un sistema de diálogo con iniciativa mixta. ATIS también fue el punto de partida de las investigaciones del MIT y de la CMU.

El grupo de investigadores del MIT ha desarrollado GALAXY, una arquitectura de sistemas de diálogo adaptable a diferentes tareas. Su finalidad es el desarrollo de sistemas de diálogo, orientados a cumplir objetivos, con posibilidad de iniciativa mixta, y aplicados a tareas reales, de uso cotidiano. Ejemplos de su arquitectura de sistemas son VOYAGER (Glass et al., 1995), sistema de información para viajeros, PEGASUS (Zue et al., 1994), sistema de acceso al sistema de reservas on-line de SABRE, o JUPITER (Zue et al., 2000), sistema de información meteorológica.

Los investigadores de la CMU han desarrollado el sistema Carnegie Mellon Communicator (CMU, 1999), (Rudnicky et al., 1999b), que permite obtener información de itinerarios complejos que incluyen múltiples reservas en vuelos, hoteles y alquiler de coches. El sistema gestiona el diálogo usando planificaciones (*schema*) y agendas. Entre sus objetivos se encuentra también facilitar la migración del sistema a otras tareas, separando en módulos agentes de dominio la gestión de la información más específica.

Investigadores de la Universidad de Colorado han desarrollado el sistema CU Communicator (Ward y Pellom, 1999), que aborda la misma tarea de acceso a información sobre vuelos, hoteles y alquiler de coches.

Estos sistemas de AT&T, MIT y CMU se encuentran operativos lo que ha permitido adquirir grandes corpus de diálogos que posibiliten ulteriores mejoras.

En el ámbito europeo, tuvo carácter pionero el proyecto SUNDIAL (*Speech Understanding and Dialogue*), relativo a la consulta de horarios de trenes y aviones, y que fue desarrollado para cuatro idiomas: inglés y francés, para la información sobre vuelos, alemán e italiano, para la información sobre horarios de trenes (Peckham, 1993).

De los diversos proyectos que siguieron, sobresalen los sistemas desarrollados por el grupo del LIMSI: el sistema PARIS-SITI (*Système d'Informations Touristiques Interactif*), dedicado a facilitar información turística sobre la capital francesa, el sistema MASK (*Multimodal Multimedia Service Kiosk*), y el ARISE (Lamel et al., 2000a), sistema de consulta telefónica sobre los trenes de la SNCF, socio en el proyecto.

Tarea muy similar a la del sistema ARISE es la del proyecto BASURDE (Bonafonte et al., 2000), desarrollado por varias universidades españolas, entre ellas la UPV, y que, actualmente, tiene continuación con el proyecto DIHANA (Benedí et al., 2004), también con participación de la UPV.

Por último, en esta breve introducción al estado del arte, cabe citar el desarrollo de un lenguaje estándar para el desarrollo de los sistemas de diálogo, llamado VoiceXML (McGlashan et al., 2004). La investigación en sistemas de diálogo ha coincidido en el tiempo con el desarrollo de la red Internet y las aplicaciones Web y, en consecuencia, se ha planteado el acceso a los sistemas de diálogo mediante aplicaciones Web. El lenguaje estándar VoiceXML es el resultado del trabajo conjunto de varias importantes compañías (AT&T, IBM, Lucent, Motorola, etc). VoiceXML ha sido diseñado para crear diálogos con audio, posibilitando la síntesis, el reconocimiento y la grabación del habla, así como el desarrollo de conversaciones con iniciativa mixta. A todas estas funcionalidades, VoiceXML aporta las ventajas del desarrollo basado en las aplicaciones Web.

2.1.1 Aplicación de métodos basados en reglas

Los investigadores de varios centros y universidades europeas desarrollaron el sistema ARISE (*Automatic Railway Information Systems for Europe*) con los objetivos de alcanzar tasas elevadas de éxito en los diálogos y de permitir que la interacción con los usuarios fuera lo más abierta y natural posible. La arquitectura del sistema (Lamel et al., 2000a) es modular, y consta de los habituales seis módulos. De los trabajos desarrollados en este proyecto, han surgido 4 prototipos: uno holandés (ARISE-NL), dos franceses (ARISE-FR1, con tecnología Philips, y ARISE-FR2, con tecnología del LIMSI) y uno italiano (ARISE-IT, con tecnología del CSELT), según los acrónimos usados en (Baggia et al., 1999).

De la investigación y prototipo del LIMSI se informa ampliamente en (Lamel et al., 2000a).

En el módulo de comprensión (Lamel et al., 2000a) se realiza, primero, un preprocesado de la cadena reconocida, aplicando reglas genéricas, reglas dependientes de la tarea y reglas dependientes del idioma, de manera que se normalizan las formas léxicas y se identifican y etiquetan conceptos. A continuación, se realiza una comprensión literal, que ya genera un frame o representación semántica, en la que se evita toda decisión que pueda llevar a malentendidos. Sin embargo, el proceso se completa con una fase de comprensión contextual, en la que se tiene en cuenta el sentido común y el conocimiento de la tarea, que permite precisar el valor de algunos atributos. Las ambigüedades que no pueda resolver el módulo de comprensión deberán ser tratadas por el módulo gestor de diálogo.

El gestor de diálogo (Lamel et al., 2000a), entonces, preguntará al usuario por datos, necesarios y aún no facilitados, o bien usará el frame dado por comprensión, junto con una serie

de reglas que gestionan la historia del diálogo, para generar una consulta SQL a la base de datos. En ambos casos, el gestor de diálogo produce un acto de diálogo en forma de frame. Y el módulo generador de respuestas lo convertirá en una frase en lenguaje natural.

El módulo más importante del sistema es el gestor de diálogo, puesto que gestiona la comprensión contextual, la historia del diálogo, la recuperación de la información (de la BD) y la generación de la respuesta. En dicho módulo se fija la estrategia de diálogo.

Por estrategia de diálogo se entiende el conjunto de decisiones tomadas por el gestor de diálogo en cualquier momento del diálogo. Los objetivos principales de la estrategia implementada en el sistema ARISE son:

- ❑ Evitar que el usuario se sienta perdido. Para ello, debe estar informado de lo que el sistema haya entendido, y así podrá corregir al sistema cuando corresponda. La confirmación implícita es el mecanismo habitual para este objetivo.
- ❑ Responder directamente a las preguntas del usuario. Del análisis de los diálogos, se desprende la conveniencia de respuestas cortas y ajustadas a la petición del usuario.
- ❑ Explicar las respuestas del sistema, si son inesperadas. Por ejemplo, aclarar la necesidad de relajar las restricciones impuestas por el usuario a fin de proporcionar la respuesta más cercana, aunque no exacta, a su petición.
- ❑ Permitir al usuario que corrija al sistema (posibilidad de negociación). Característica que posibilita enmendar errores del reconocedor y también errores, o cambios de idea, del propio usuario.
- ❑ Evitar los malentendidos. Para esto, es esencial reducir los errores de reconocimiento. Resultó conveniente rechazar las palabras identificadas con baja fiabilidad.

Se procuró que estos objetivos fueran compatibles con una estrategia de diálogo mixta. El usuario puede preguntar por los diferentes tipos de información disponibles sin necesidad de comandos específicos o una navegación prefijada. El sistema permite la negociación, esto es, el usuario puede rechazar la información suministrada y, en tal caso, modificar, relajar o restringir los criterios o datos para la búsqueda. Al existir la negociación, el sistema informará sólo de un tren (el más próximo a las condiciones de búsqueda) y quedará en disposición de, si hay negociación, informar de otro.

Sin embargo, la estrategia finalmente implementada en ARISE no es siempre de iniciativa mixta. Se diseñó una estrategia en dos niveles. De entrada, se activa la estrategia de iniciativa mixta, de manera que el usuario puede plantear cualquier consulta. Ahora bien, cuando se detectan problemas (particularmente, en la obtención de los valores de los atributos fundamentales de la consulta: origen, destino y fecha del viaje), se pasa a una estrategia dirigida por el sistema. En tales casos, se arrancan subdiálogos restringidos a preguntar o confirmar uno de esos atributos. El grado de control por parte del sistema se incrementa conforme el usuario se retrase en aportar el dato solicitado, llegando, en el caso de las ciudades, a pedirle que éste sea deletreado.

2.1.2 Aplicación de métodos estocásticos

Respecto a la aplicación de métodos estocásticos en el desarrollo de sistemas de diálogo cabe citar, entre otras, (Levin et al., 2000a), (Oh y Rudnicky, 2002), (Stolcke et al., 2000) y (Williams y Young, 2003).

- AT&T (Levin et al., 2000)

En (Levin et al., 1998), (Levin et al., 1999) y (Levin et al., 2000a), los investigadores de AT&T exponen cómo el problema de la gestión del diálogo puede analizarse como un problema de optimización de una función objetivo, representativa de las dimensiones relevantes en la tarea de diálogo considerada. Este planteamiento lleva a considerar el sistema de diálogo mediante un modelo estocástico, el MDP (proceso de decisión de Markov), y la posibilidad de aprendizaje automático de las estrategias de diálogo más adecuadas mediante la interacción con un usuario simulado.

Dada la naturaleza dinámica del diálogo, los investigadores de AT&T consideran que no es adecuado un aprendizaje supervisado de las estrategias a partir de un corpus, en particular por las dificultades de evaluar el sistema cuando los diálogos ejecutados se desvíen de los registrados en el corpus. La propuesta de (Levin et al., 1998) y (Levin et al., 2000a) es el aprendizaje de los modelos mediante una combinación de:

- ❑ Aprendizaje reforzado (*reinforcement learning*). El sistema aprenderá la estrategia óptima en su interacción con los usuarios. Esta técnica permite explorar el espacio de estados y, periódicamente, actualizar los valores de todos los estados visitados.
- ❑ Aprendizaje supervisado. Para aplicar razonablemente la anterior técnica, es muy recomendable disponer de un usuario simulado. El aprendizaje supervisado a partir del corpus permite estimar los parámetros de tal usuario.

El diálogo puede formalizarse, (Levin et al., 1998), (Levin et al., 1999) y (Levin et al., 2000a), como un proceso de decisión secuencial en términos de:

- ❑ Un conjunto de acciones (*action set*). Incluye todas las posibles acciones del sistema: interacciones con el usuario, interacciones con recursos externos, procesamiento interno.
- ❑ Un espacio de estados (*state space*). Un estado representa todo el conocimiento que el sistema tiene acerca de los recursos internos y externos con los que interactúa. Una sesión de diálogo es un camino desde el estado inicial al estado final de este espacio.
- ❑ Una estrategia. En cada estado alcanzado, la estrategia especifica la siguiente acción a realizar. Establece, por tanto, una correspondencia (*mapping*) entre el espacio de estados y el conjunto de acciones. La estrategia óptima minimizará una función objetivo, cuyos términos serán los costes esperados para las diferentes dimensiones del diálogo.

Si N_a = número de acciones y N_s = número de estados, entonces el número de posibles estrategias es exponencial: $N_a^{N_s}$, y no se pueden explorar exhaustivamente. Para llevar a cabo el aprendizaje automático, hay que considerar el proceso de decisión como un MDP, considerando las siguientes aproximaciones para las probabilidades de las transiciones y para los costes:

$$P(s_{t+1} | s_t, \dots, s_0, a_t, \dots, a_0) = P_T(s_{t+1} | s_t, a_t)$$

$$P(c_t | s_t, \dots, s_0, a_t, \dots, a_0) = P_C(c_t | s_t, a_t)$$

El coste óptimo de un estado s será la suma esperada de los costes desde ese estado hasta el estado final s_F siguiendo la estrategia óptima π^* :

$$V^*(s) = \langle \sum_{t=0..T_F} c(s_t, a_t) \rangle$$

El valor óptimo de la función en un estado s_t se puede calcular como suma del coste esperado instantáneo en s_t y el coste esperado del siguiente estado s_{t+1} usando la mejor acción disponible:

$$V^*(s_t) = \min_a [\langle c(s_t, a) \rangle + \sum_s P_T(s_{t+1} = s | s_t, a) V^*(s)]$$

Y la estrategia óptima se obtiene de:

$$\pi^*(s_t) = \arg \min_a [\langle c(s_t, a) \rangle + \sum_s P_T(s_{t+1} = s | s_t, a) V^*(s)]$$

En (Levin et al., 1998) y (Levin et al., 2000a) se aplica esta técnica para obtener automáticamente una estrategia óptima para la tarea ATIS. Los resultados experimentales muestran que el algoritmo converge, alcanzando la estrategia que se juzga óptima al cabo de 710.000 diálogos.

En (Pieraccini et al., 1997), el grupo investigador de AT&T propone, con AMICA, una arquitectura genérica que permita un rápido desarrollo de sistemas de diálogo hablado, con capacidad de iniciativa mixta, para diferentes tareas. Como ideas principales, se destaca que la estrategia es bastante independiente de las aplicaciones concretas y que el sistema se puede descomponer en módulos que realicen distintas acciones o funciones. A éstas se las denomina acciones de diálogo (*dialogue actions*).

En (Levin et al., 1999), se concreta algo más la descripción de la arquitectura AMICA. Los estados del diálogo (*state space*) se representan mediante plantillas (*templates*), estructuras recursivas de pares clave / valor. Las acciones (*action set*) son funciones que leen una plantilla, pueden interactuar con el entorno, y devuelven una plantilla modificada. Sería conveniente disponer de un conjunto pequeño de acciones de diálogo de alto nivel, genéricas. La experiencia, sin embargo, indica que no es posible una adecuada parametrización de tales acciones de alto nivel con independencia de la aplicación o tarea. La solución consiste en implementar un conjunto de acciones elementales, que serán los bloques para construir acciones más complejas, mediante un lenguaje de scripts.

DMD (*dialogue manager developer*) es el lenguaje de scripts propuesto, tanto para construir las acciones de diálogo como la estrategia del diálogo. Acciones y estrategia se codifican en una base de datos de scripts DMD. El módulo gestor de diálogo, entonces, se convierte en un intérprete de DMD. Según la arquitectura del SDS, el gestor de diálogo

puede mantener el estado de diálogo (*stateful mode*) o carecer de estado (*stateless mode*) y recibirlo de otro módulo, como el *hub* en la arquitectura MIT Galaxy. Una de las ventajas de DMD es soportar estos dos modos de operación.

En (Levin et al., 2000b), se describe el diseño e implementación del AT&T DARPA Communicator. La arquitectura del SDS está centralizada en un *hub*, que funciona como enrutador programable que invoca los diferentes servidores del sistema y retransmite los mensajes (*Galaxy frames*) que intercambian.

- University of Cambridge (Young et al., 2002)

En (Williams y Young, 2003), investigadores de la Universidad de Cambridge consideran también la aproximación markoviana, el MDP aplicado al sistema de diálogo, y el uso del aprendizaje reforzado (*reinforcement learning*) para que el gestor de diálogo aprenda la estrategia más adecuada. En particular, la cuestión objeto de su estudio es cómo crear el sistema inicial (*bootstrapping*).

Para abordar este problema, el método que plantean, sucintamente, consiste en:

- ❑ Realizar, grabar y transcribir sesiones de diálogo usando la técnica del mago de Oz.
- ❑ Delimitar las tareas propias del dominio y excluir del corpus adquirido los turnos de fuera del dominio (*out-of-domain turns*).
- ❑ Enumerar el conjunto de acciones y el espacio de estados (*action set* y *state space*, componentes del MDP), anotando los turnos con actos de diálogo. Dado el inmenso tamaño del espacio de estados, se propone usar una función de valor parametrizada que agruparía estados y también usar un algoritmo de aprendizaje supervisado para estimar esta función.
- ❑ Establecer una estrategia (*policy*) inicial, es decir, la correspondencia entre estados y acciones, en términos del MDP. Para ello, tomando las acciones como variable, se usa aprendizaje supervisado para construir un clasificador que fije esa primera estrategia.

El método se ha probado usando el corpus Autoroute, que consta de 166 diálogos, en los que el mago de Oz atiende consultas sobre direcciones y rutas en el Reino Unido. En la discusión de los resultados, se indica que basta una cantidad pequeña de diálogos (en torno a 30 ó 40 diálogos, con varios centenares de turnos) para disponer de la mayor parte de la información relevante para los algoritmos de aprendizaje supervisado. Los modelos así obtenidos permiten predecir correctamente la acción del mago en un 72.9% de los turnos, frente a sólo un 58.4% si se usa una estrategia diseñada a mano.

Sin embargo, también se aprecia que la mitad de los turnos corresponden a estados que sólo se visitan una vez. Esto confirma la necesidad de cantidades masivas de datos para poder observar adecuadamente todos los estados de diálogo válidos.

Más recientemente, en (Williams et al., 2005a – 2005b), los mismos investigadores mejoran sus resultados mediante el uso de un MDP parcialmente observable (POMDP). Para resolver un problema como el de la gestión de diálogo, caracterizado por la planificación de

acciones en condiciones de incertidumbre, parece más adecuado un POMDP que un MDP completamente observable.

Un POMDP se define mediante una tupla $\{S, A_m, T, R, O, Z\}$, donde S es un conjunto de estados, A_m es un conjunto de acciones (del sistema), T define una probabilidad de transición, $p(s' | s, a_m)$, R define la recompensa esperada, $r(s, a_m)$, O es un conjunto de observaciones y Z define una probabilidad de observación, $p(o' | s', a_m)$. Las transiciones en un POMDP llevan a estados no observados, supuestos (“creencia del sistema”, *belief state*) en función de cierta evidencia que dan las observaciones. Según esta propuesta, se puede calcular b , “belief state”, mediante:

$$b'(s') = k \cdot p(o' | s', a_m) \cdot \sum_{s \in S} p(s | a_m, s) \cdot b(s)$$

La propuesta de (Williams et al., 2005a – 2005b) es descomponer la observación en dos componentes: un componente discreto, h , que representa la hipótesis del reconocedor del habla, y un componente continuo, c , que representa la medida de confianza. De este modo, las medidas de confianza se integran en el modelo de diálogo representado mediante un POMDP.

En los resultados experimentales, el modelo POMDP supera a los MDP tradicionales. Además, el método descrito permite mejorar automáticamente el funcionamiento de gestores de diálogo sencillos (diseñados a mano) mediante la monitorización del *belief state* usando la información de las medidas de confianza.

Como trabajo futuro, se considera incluir la lista de N-mejores hipótesis del ASR (en lugar de una hipótesis) y ampliar la escala del modelo para tareas reales (en el artículo se trata una tarea muy simplificada).

2.2 Comprensión del lenguaje

En esta sección se revisan algunas aportaciones relevantes en el área de la comprensión del lenguaje natural, especialmente desde una aproximación estocástica. Los trabajos que se comentan son los siguientes:

AT&T (Levin y Pieraccini, 1995)

LIMSI (Minker et al., 1996)

Radboud University Nijmegen (Sturm et al., 1999)

CMU (Rudnicky et al., 1999)

Philips (Sougvinier et al., 2000)

MIT (Zue et al., 2000)

Universidad Politécnica de Valencia (Segarra et al., 2002)

University of Cambridge (Young, 2002)

Aachen University of Technology (Macherey et al., 2003)

Université d'Avignon (Raymond et al., 2003)

AT&T (Potamianos et al., 2005)

- *AT&T (Levin y Pieraccini, 1995)*

En (Levin y Pieraccini, 1995), se describe CHRONUS, un módulo de comprensión diseñado para la tarea ATIS. CHRONUS es un módulo híbrido, que usa un modelo estocástico, aprendido a partir de un corpus anotado, y otros componentes de conocimiento, diseñados a mano (reglas lingüísticas). La comprensión se lleva a cabo en dos fases, realizadas por dos componentes de CHRONUS.

El primer componente es el módulo de análisis local, que recibe las frases en lenguaje natural, efectúa su preprocesado léxico y segmentación conceptual, y genera una(s) plantilla(s), representación semántica de la entrada recibida. El segundo componente es el módulo intérprete y se encarga de resolver las ambigüedades que hayan quedado después del análisis local (es decir, este intérprete modifica las plantillas que recibe teniendo en cuenta la información de contexto, de toda la intervención del usuario así como de anteriores intervenciones).

Es en el primer componente, el módulo de análisis local, donde los investigadores de AT&T usan modelos estocásticos, en concreto modelos ocultos de Markov (HMMs) cuyos estados corresponden a los diferentes conceptos de la tarea. El análisis local consiste en una búsqueda de Viterbi que genera la segmentación más probable de la frase de entrada en segmentos etiquetados mediante conceptos.

- LIMSI (Minker et al., 1996)

En (Minker et al., 1996), los investigadores del LIMSI abordan la aproximación estocástica a la comprensión del lenguaje y comparan los resultados con los obtenidos anteriormente usando una aproximación basada en reglas (mediante una gramática de casos, *a rule-based case grammar*). El modelo estocástico considerado es un modelo oculto de Markov de primer orden. Consideran la tarea ARPA ATIS para entrenar el modelo y realizar la comparativa de resultados entre ambas aproximaciones.

En el aprendizaje, se estiman los parámetros del modelo (un *bigram backoff HMM*) teniendo en cuenta las secuencias de palabras preprocesadas (las observaciones) y sus correspondientes secuencias semánticas (*semantic sequences*) que definen los estados del modelo. Para llevar a cabo este aprendizaje, se han usado 6.439 consultas de los corpus de ATIS. Se obtiene un modelo de 330 estados (diferentes secuencias semánticas).

La tarea de comprensión se descompone en dos subtareas realizadas por los siguientes módulos: un preprocesador de conceptos (*conceptual preprocessor*) y un analizador semántico (*semantic decoder*). El preprocesador de conceptos realiza un análisis morfológico (tratamiento de expresiones compuestas, unificación de formas flexionadas, agrupamiento en clases de palabras de significado parecido, identificación de palabras no relevantes y de palabras fuera del dominio, etc). Este análisis reduce el número de entidades léxicas de 1.164 a 737, lo que simplifica bastante la siguiente subtask. El analizador semántico, usando el HMM aprendido a partir del corpus, genera la secuencia semántica más probable dada la consulta preprocesada que recibe en su entrada.

En la evaluación, se aprecia que la aproximación basada en reglas obtiene mejores resultados, aunque esto era previsible dado que el modelo estocástico considerado es bastante simple. Al tratarse de un modelo de bigramas, no se tiene en cuenta la información del contexto. En trabajo futuro, se plantea la necesidad de incluir esta información de contexto en el modelo para mejorar el rendimiento.

Por otra parte, se destaca que la aproximación estocástica es más flexible ante cambios en la tarea o el idioma. Mientras que la aproximación basada en reglas requiere el diseño y mantenimiento de una gramática de casos, en la aproximación estocástica basta con la tarea de etiquetar un nuevo corpus (tarea más simple que diseñar a mano una gramática) y, a partir del corpus etiquetado entrenar, automáticamente, los modelos.

En (Minker, 1997) se comprueba esto último, al presentar el desarrollo del mismo módulo de comprensión adaptado de la tarea ATIS (tarea sobre consulta de vuelos, en inglés) a la tarea MASK (tarea sobre información turística, en francés). La evaluación muestra que, en la tarea MASK, la aproximación estocástica alcanza menores tasas de error en la identificación de las secuencias semánticas que la aproximación basada en reglas.

- DSIC-UPV (Segarra et al., 2002)

En (Segarra et al., 2002) se aborda la comprensión del lenguaje desde una aproximación estocástica y considerando el problema desde el punto de vista de una traducción. Así, el módulo de comprensión recibe secuencias de palabras (en el caso de un sistema de diálogo,

frases o intervenciones del usuario) y genera *frames* (unidades semánticas que describen el significado de las intervenciones del usuario, en el caso de los diálogos). Y esta traducción de frases a *frames* se realiza mediante traductores (*transducers*) aprendidos automáticamente.

La tarea de comprensión se descompone en dos fases. En la primera fase, la frase de entrada, en lenguaje natural, se traduce en una frase semántica, definida en un lenguaje semántico intermedio (*intermediate semantic language*, ISL) que es secuencial. En la segunda fase, dicha frase semántica se traduce en el *frame*, o *frames*, que corresponda. En la traducción a ISL se aplican técnicas de aprendizaje automático (transductores secuenciales), y en la traducción a *frames* se aplica un sencillo sistema basado en reglas.

La secuencialidad del ISL respecto al lenguaje natural, permite realizar una segmentación de la frase de entrada en intervalos que corresponderán a las unidades semánticas del ISL. De este modo, y si W es el vocabulario de la tarea y V es el alfabeto de unidades semánticas, el corpus de entrenamiento será un conjunto de pares (u, v) donde: $v = v_1v_2 \dots v_n$, $v_i \in V$, $i = 1 \dots n$, $u = u_1u_2 \dots u_n$, $u_i = w_{i1}w_{i2} \dots w_{i|u_i|}$, $w_{ij} \in W$, $i = 1, \dots, n$, $j = 1, \dots, |u_i|$.

En el aprendizaje del ISL, se han obtenido dos tipos de modelos: un modelo para el lenguaje semántico $L_s \subseteq V^*$, y un conjunto de modelos, uno para cada unidad semántica, $v_i \in V$. Ambos tipos de modelos han de ser autómatas de estados finitos estocásticos.

En la experimentación, se observa que usando modelos de bigramas se alcanzan menores tasas de error (tanto en identificación de unidades semánticas como de frames) que si se usan otras técnicas basadas en inferencia gramatical, como la MGGI.

- (Potamianos et al., 2005)

En los sistemas de diálogo hablado, según (Potamianos et al., 2005), la comprensión del lenguaje se encuentra estrechamente vinculada con la determinación del estado de diálogo. En consecuencia, estos autores formulan el problema de la comprensión como la determinación del estado de diálogo S_t dada la entrada de usuario W_t en términos de secuencias de palabras. Ahora bien, como incluyen la decodificación acústica como fase previa de la comprensión, consideran que la entrada observable es O_t , la secuencia de vectores acústicos, mientras que S_t y W_t son variables ocultas y la comprensión consiste en una búsqueda conjunta en el espacio de estados W_t, S_t .

Por tanto, y según la aproximación de máxima verosimilitud, hay que maximizar la distribución de probabilidad a posteriori conjunta:

$$\max_{S_t, W_t} P(S_t, W_t | O_t, S_1 \dots S_{t-1}) = \max_{S_t, W_t} P(O_t | W_t) \cdot P(W_t | S_1 \dots S_t) \cdot P(S_t | O_t, S_1 \dots S_{t-1})$$

Donde S_t es el estado de diálogo, W_t es la secuencia de palabras del usuario y O_t es la secuencia acústica observada, en el turno de diálogo t .

En lugar de realizar esta maximización conjunta, se suele, primero, maximizar la probabilidad a posteriori respecto a W_t y, después, respecto a S_t :

$$W'_t = \arg \max_{W_t} P(O_t | W_t) \cdot P(W_t | S_1 \dots S_{t-1})$$

$$S'_t = \arg \max_{S_t} P(W'_t | S_1 \dots S_t) \cdot P(S_t | S_1 \dots S_{t-1})$$

La segunda maximización corresponde, propiamente, al problema de la comprensión. Las probabilidades se estiman mediante modelos de comprensión λ_U y de diálogo λ_D , de manera que el problema se formula como:

$$S'_t = \arg \max_{S_t} P(W'_t | S_t, \lambda_U)^{\gamma_U} \cdot P(S_t | S_{t-1}, \lambda_D)^{\gamma_D}$$

Donde γ_U y γ_D son pesos dependientes de la tarea y que pueden variar con el tiempo.

Los modelos de comprensión λ_U considerados son modelos de n-gramas (VNSA: *Variable Ngram Stochastic Automaton*, modelos ya referenciados en anteriores trabajos como Riccardi y Gorin, 2000). Igualmente, los modelos de diálogo λ_D son modelos de n-gramas, considerándose suficiente los bigramas (es decir, $P(S_t | S_1 \dots S_{t-1}) \approx P(S_t | S_{t-1})$) para representar las dependencias entre estados en diálogos con iniciativa del usuario.

En la experimentación, se informa de que la incorporación de los modelos de diálogo λ_D permiten mejorar los resultados de comprensión entre un 5% y un 15%, en exactitud en el etiquetado de estado de diálogo (LACC: *dialogue state label accuracy*).

- University of Cambridge (Young, 2002)

También en (Young, 2002) la comprensión del lenguaje se plantea como la obtención de un conjunto de actos de diálogo, $A_u = \{a_1, a_2, \dots\}$, a partir de la señal acústica, $Y = Y_i$, y dado el estado actual del sistema, S_s :

$$A'_u = \arg \max_{A_u} P(A_u | Y, S_s) = \arg \max_{A_u} P(Y | A_u, S_s) \cdot P(A_u | S_s)$$

Igualmente, el proceso de comprensión se descompone en dos fases sucesivas, en donde la secuencia de palabras, W , constituye el punto intermedio:

$$A'_u = \arg \max_{A_u} \{ \arg \max_W \{ P(Y | W) \cdot P(W | S_s) \cdot P(A_u | W, S_s) \} \}$$

$$W' = \arg \max_W \{ P(Y | W) \cdot P(W | S_s) \}$$

$$A'_u = \arg \max_{A_u} \{ P(A_u | W', S_s) \}$$

Como antes, el problema de la comprensión del lenguaje se asocia a la segunda maximización. Ésta, a su vez, se suele llevar a cabo introduciendo una representación intermedia, mediante conceptos semánticos, C :

$$A'_u = \arg \max_{A_u} \{ \arg \max_C \{ P(A_u | C, S_s) \cdot P(C | W', S_s) \} \}$$

$$C' = \arg \max_C \{ P(C | W', S_s) \}$$

$$A'_u = \arg \max_{A_u} \{ P(A_u | C', S_s) \}$$

La primera ecuación representa el proceso de la extracción semántica, y es habitualmente resuelta usando HMMs. Otras alternativas son los modelos de gramáticas libres de contexto estocásticas (SCFG), y otras variantes de HMM: los modelos de Markov de vector de estados y los modelos de Markov de vector de estados oculto.

La segunda ecuación representa el proceso de combinar el estado del sistema y las representaciones semánticas del usuario para encontrar los actos de diálogo más probable. Para ello, se acostumbra a usar redes bayesianas.

- Radboud University Nijmegen (Sturm et al., 1999)

En (Sturm et al., 1999a), el módulo de comprensión (NLP-module) del prototipo ARISE-NL recibe las frases reconocidas en forma de grafo de palabras y, usando modelos estocásticos (de unigramas y bigramas de conceptos), determina la frase más verosímil así como su significado.

En (Bouwman et al., 1999), se expone cómo se incorporan medidas de confianza en el módulo de comprensión, en ARISE-NL. El grafo de palabras representa todas las hipótesis de palabras de una intervención (*utterance*) dada. En comprensión, una gramática específica de la tarea identifica los conceptos que corresponden a secuencias de palabras significativas. Se construye así su correspondiente grafo de conceptos. Cada concepto permite obtener, al menos, un atributo (el elemento más simple de información).

Cada frase tendrá una probabilidad calculada a partir de las marcas de confianza acústicas de sus palabras (dadas por el reconocedor), la probabilidad del modelo de lenguaje y la probabilidad de la gramática de conceptos. Para determinar la fiabilidad de un atributo, se consideran las frases dentro de un rango predeterminado respecto a la más probable. Si en todas estas frases aparece cierto atributo, éste será fiable, y, en caso contrario, se marcará como inseguro.

El parámetro más importante en esta técnica es el umbral que establece el rango de frases a considerar. Si el rango es demasiado amplio, la mayoría de las frases correctas tendrán competidores y, en consecuencia, aumentará el número de atributos correctamente reconocidos pero marcados como no fiables (*false rejection rate*). Si el rango es muy pequeño, se aceptarán hipótesis incorrectas y, por tanto, atributos incorrectamente reconocidos serían marcados como fiables (*false acceptance rate*). El valor óptimo ha sido determinado experimentalmente.

Este uso de las medidas de confianza en comprensión repercute en la funcionalidad del SDS, dada la selección de confirmaciones condicionada a la confianza de los atributos que se lleva a cabo en el gestor de diálogo, (Sturm et al., 1999a) y (Sturm et al., 1999b).

- Aachen University of Technology (Macherey et al., 2003)

En (Macherey et al., 2003a), donde se presenta un mecanismo de detección de errores en reconocimiento (ASR) y de comprensión (NLU) y su uso en gestión del diálogo, se trata brevemente la aproximación seguida en el desarrollo del módulo NLU. Dada la transcripción de la señal reconocida por el ASR, el módulo NLU determina la secuencia de conceptos más probable según los modelos utilizados. Se aplica una metodología basada en máxima entropía (ME) para abordar la tarea de comprensión. Usando el algoritmo GIS (*Generalized Iterative Scaling*), se estiman los parámetros asociados a cada función característica que se considera en el modelo de ME. Las medidas de confianza facilitadas por el ASR constituyen una función característica más a integrar en el cómputo del modelo de ME.

- Université d'Avignon (Raymond et al., 2003)

En (Raymond et al., 2003) y (Raymond et al., 2004), se cuestiona el reconocimiento (ASR) basado en listas *N-best* generadas por enumeración de los mejores N caminos en los grafos de palabras. Estas listas se suelen ordenar de acuerdo a una combinación de indicadores acústicos y de modelo de lenguaje, sin implicar otros niveles lingüísticos. Las diferencias entre las hipótesis acostumbran a ser pequeñas y pueden estar originadas en ruidos o efectos del habla espontánea. Esto supone una gran debilidad desde la perspectiva del sistema en su conjunto (y del gestor de diálogo, en particular), pues no todas las palabras tienen la misma importancia y todas las hipótesis que difieran sólo por los efectos del habla espontánea pueden considerarse equivalentes.

La alternativa planteada por este grupo de investigadores de la Universidad de Avignon es generar, además de la lista de las mejores N hipótesis de palabras, otra lista con las mejores N interpretaciones. Cada interpretación reflejaría un significado distinto, desde el punto de vista del gestor de diálogo, es decir, estaría asociada a un determinado estado de diálogo, historia del diálogo, o creencia del sistema (*system belief*). Las interpretaciones constan de secuencias de conceptos, deducidas de modelos conceptuales (SFST, *Stochastic Finite State Transducers*, propios del gestor de diálogo), que se aplican a los grafos de palabras reconocidas.

La lista de las mejores N hipótesis de palabras se estructuraría entonces de acuerdo a las distintas interpretaciones que se encontrara. Las interpretaciones se ordenarían por el indicador de su primera hipótesis de palabras; y las hipótesis de palabras, correspondientes a una interpretación dada, simplemente por sus indicadores (*scores*).

El objetivo de la arquitectura propuesta es facilitar al gestor de diálogo un número reducido de hipótesis, pero que sean diferentes en su significado y estén validadas por medidas de confianza. Se ha realizado una evaluación con un corpus de diálogos de consultas turísticas de France Telecom, y se informa de reducciones de los errores de comprensión (significativas en *concept error rate* y *understanding error rate*, y moderadas en *word error rate*, como se podía esperar dada la naturaleza de la nueva fuente de información considerada).

En (Raymond et al., 2004), además, se propone una nueva aplicación para el aprendizaje automático de los árboles de decisión, que permiten deducir las interpretaciones, considerando las confianzas acústicas de las palabras que expresan los conceptos, así como la consistencia lingüística y semántica de las hipótesis. Este aprendizaje posibilita una mejora muy significativa en términos de *understanding error rate*.

En (Béchet et al., 2004), siguiendo la misma línea de investigación, se trata en más detalle el módulo de comprensión. Se destaca la vinculación más estrecha entre los módulos de reconocimiento y comprensión, que permite mejorar la exactitud del reconocimiento y proporcionar mejores interpretaciones para uso del gestor de diálogo.

- MIT (Zue et al., 2000)

En (Zue et al., 2000), se expone cómo funciona TINA, módulo de comprensión desarrollado por el grupo de investigadores del MIT, integrado en el sistema JUPITER. Este analizador transforma secuencias de palabras en representaciones semánticas o frames, y, en JUPITER, actúa en dos situaciones:

- ❑ Comprensión del usuario (*query understanding*). Procesa las intervenciones del usuario, seleccionando las mejores hipótesis del grafo de palabras dado por el ASR. Utiliza una gramática, diseñada a mano, que codifica información sintáctica y semántica. En la fase final del proceso, se consideran los valores (*scores*) asociados al reconocimiento y calculados por el analizador, y así como el contexto previo del diálogo.
- ❑ Comprensión de la base de datos (*content understanding*). De modo automático, tres veces al día, los informes meteorológicos (textos redactados por los expertos del National Weather Service) se traducen a frames.

- CMU (Rudnicky et al., 1999)

En (Rudnicky et al., 1999b), la comprensión se realiza con un analizador semántico Phoenix, que usa una gramática derivada del sistema CMU ATIS, aumentada con transcripciones recopiladas del uso del propio prototipo (CMU Communicator). La correspondencia entre las frases del usuario y los conceptos (*topic*) se establece mediante un conjunto de redes receptoras (*receptor nets*) asociadas con cada *schema*.

En (Rudnicky y Xu, 1999a), este conjunto de conceptos o redes, resultado del analizador Phoenix, se correlacionará con manejadores (*handlers*) de una agenda, dado que, en este papel, se sustituye la aproximación al gestor de diálogo basada en *scripts* por otra, más dinámica, basada en una agenda.

- Philips (Sougvinier et al., 2000)

En (Sougvinier et al., 2000), los investigadores de Philips describen su módulo de comprensión. Éste recibe del ASR un grafo de palabras y lo interpreta valiéndose de tres componentes principales:

- Una gramática libre de contexto estocástica, que, dada una hipótesis de frase, $W = (w_1, \dots, w_n)$, proporciona una secuencia de segmentos, asociados a conceptos o a frases de relleno, (c_1, \dots, c_N) , donde cada segmento c_i está asociado a una secuencia de palabras (w_j, \dots, w_k) de la frase W , y se identifica con un concepto C_i con una cierta probabilidad, $P(c_i|C_i)$.
- Un modelo de lenguaje de conceptos (*concept LM*), que proporciona probabilidades a las secuencias de conceptos. Como el número de conceptos en una frase suele ser pequeño, es suficiente un modelo de bigramas.

$$P(W) = P(C_{N+1}|C_N) \cdot \prod_{i=1 \dots N} P(c_i|C_i) \cdot P(C_i|C_{i-1})$$

- Un modelo de lenguaje de relleno (*filler LM*), que cubre las partes de la intervención que no puedan analizarse. Se usa un modelo de bigramas a nivel de palabras.

La posibilidad de un análisis parcial, con segmentos significativos y otros de relleno, proporciona la robustez necesaria para tratar los problemas del habla espontánea.

Ante la habitual dificultad de la falta de corpus de tamaño suficiente para aprender los modelos, en (Sougvinier et al., 2000) se discuten varias posibilidades:

- Uso de modelos categorizados, que, en el caso de bigramas, permitan el cómputo: $P(w_2|w_1) = P(w_2|c(w_2)) \cdot P(c(w_2)|c(w_1))$
- Ampliación artificial del corpus de entrenamiento. Invirtiendo el proceso de análisis, se pueden obtener frases a partir de la gramática, que constituirán un corpus artificial para entrenar modelos de n-gramas. Para elegir aleatoriamente las reglas de la gramática se usan métodos Monte Carlo.
- Adaptación on-line no supervisada. Se trata de actualizar los modelos justo después de cada reconocimiento. Los resultados experimentales muestran que esta adaptación empeora los modelos si se usa sólo la mejor hipótesis del ASR. En cambio, resultó efectiva si se usaban todas las hipótesis de la lista *N-best*. El motivo es que, a diferencia de los errores de reconocimiento, las partes bien entendidas aparecen en la mayoría de las hipótesis.

Finalmente, en cuanto módulo de comprensión, en (Sougvinier et al., 2000) se apuntan algunas ideas innovadoras para mejorar su exactitud usando fuentes de información que corresponden a otros módulos “posteriores” en la arquitectura habitual de los SDS:

- Incorporar restricciones de la base de datos. Dado I , el conjunto de atributos presentes en la hipótesis, se puede calcular $n(I)$, el número de tuplas en la BD que coinciden con los valores en I . Se define la distribución: $P_{DB}(I) = \{ 1, \text{ si } n(I)>0; 0, \text{ si } n(I)=0 \}$. Y la probabilidad de la hipótesis se reajusta a $P(W) \cdot P_{DB}(I(W))$. La experimentación muestra una mejora del 21% en tasa de error de atributos.
- Aplicar pruebas de consistencia con la historia del diálogo (*system belief*). Consiste en rechazar las hipótesis que, tras combinarlas con la actual creencia del sistema, no se refieran a ninguna entrada válida en la BD. Permite detectar información contradictoria con los anteriores turnos y que, presumiblemente, corresponde a errores de reconocimiento.

2.2.1 Uso de medidas de confianza

Existe la necesidad de identificar y tratar los errores de reconocimiento puesto que los módulos ASR, a pesar de sus continuas mejoras, posiblemente siempre generen cierto número de errores que, propagados a lo largo del sistema, puedan causar el fracaso de la sesión de diálogo en que se produzcan. Un mecanismo ampliamente aceptado para tratar este problema es la asociación de distintas medidas de confianza, a distintos niveles (fonético, de palabra, de intervención completa) a las hipótesis que proporcione el ASR.

En esta sección se comentan algunos trabajos acerca del uso de medidas de confianza en la comprensión del lenguaje natural:

(Bouwman y Hulstijn, 1998)

Aachen University of Technology (Wessel et al., 1998)

University of Colorado (San-Segundo et al., 2001)

MIT (Hazen et al., 2002)

Universidad de Vigo (Pérez-Piñar y García, 2005)

- MIT (Hazen et al., 2002)

Para obtener un sistema con esta capacidad, en (Hazen et al., 2002), se plantean dos objetivos: a) desarrollar una técnica de asignación de confianzas de reconocimiento que determine exactamente qué es, o no es, fiable en las hipótesis del ASR; b) integrar las medidas de confianza en los siguientes módulos, comprensión y gestor de diálogo, para que éstos puedan tomar las decisiones correctas. Dada la orientación de la presente tesis, este segundo objetivo es de especial interés.

En (Hazen et al., 2002) se describe con amplitud cómo se integran las medidas de confianza en TINA, el módulo de comprensión que forma parte del sistema JUPITER. En TINA se analiza cada intervención usando una gramática libre de contexto estocástica y etiquetada semánticamente. Tratar las medidas de confianza en TINA supone:

- ❑ Añadir las medidas de confianza a la lista de n mejores hipótesis (*N-best list*) que proporciona el ASR.
- ❑ Colapsar la *N-best list* en un grafo de palabras, donde a los arcos se asocian las medidas de confianza a nivel de palabra. Con éstas, y con probabilidades dadas por el modelo de lenguaje, se puede calcular una puntuación para cada hipótesis.
- ❑ Buscar y seleccionar en el grafo la hipótesis con mejor puntuación y que produzca un análisis completo de la intervención. Si no existe, se buscará un análisis parcial, robusto, y con la mejor puntuación.

- ❑ Ampliar la gramática para asignar una función sintáctica o semántica a palabras rechazadas cuando el contexto lo permita. Por ejemplo, es posible que una palabra rechazada, no identificada, sea analizada como “nombre de ciudad desconocido” si por el contexto se tiene la certeza de que corresponde a alguna ciudad.

Se evaluó JUPITER, con TINA usando medidas de confianza, considerando cuatro estrategias de rechazo de hipótesis:

- ❑ Sin rechazo. La mejora respecto al funcionamiento de referencia (sin medidas de confianza) es pequeña. En ambas situaciones se produce un número excesivo de errores de inserción. Esto se debe a que, sin rechazo, se obliga a encontrar una interpretación razonable para todas las hipótesis de la lista. La estrategia es adecuada cuando el usuario es cooperativo, pero no cuando la intervención es ajena al dominio, contiene palabras desconocidas o artefactos.
- ❑ Con rechazo a nivel de intervención (*utterance-level rejection*). Los errores de inserción se reducen significativamente, a costa de un ligero incremento de los errores de borrado. Respecto a la anterior estrategia, supone una mejora del 17%.
- ❑ Con rechazo a nivel de palabra (*word-level rejection*), además del de intervención. Se consideran dos variantes: *optional word rejection*, que produce más errores de inserción que de borrado, y *hard word rejection*, donde los borrados superan a las inserciones. Aunque difieren en la naturaleza de los errores, ambas proporcionan casi la misma tasa de error. Respecto a la estrategia precedente, supone una mejora del 14% en reducción de tasa de errores.

Respecto al sistema original (*baseline*), sin medidas de confianza, se alcanza una reducción relativa del 35% en tasa de errores en conceptos (*concept error rate*).

En cuanto al uso de las medidas de confianza en el gestor de diálogo, en (Hazen et al., 2002) se comenta su incidencia en dos acciones: la selección de hipótesis y la planificación de las respuestas.

El gestor de diálogo puede valerse de las medidas de confianza, así como de su conocimiento de reglas pragmáticas subyacentes en la tarea, para determinar la fiabilidad de los datos presentes en el frame recibido. Así, si el frame contiene información contradictoria (por ejemplo, los nombres de una ciudad y de un estado que no son compatibles), puede etiquetar como inválido el dato con menor confianza y preguntar a continuación por éste (por ejemplo, si el dato inseguro es la ciudad, suministrando al usuario una lista de las ciudades que el sistema conoce en ese estado del país).

El ejemplo anterior lo es también de la planificación de respuestas condicionada a las medidas de confianza. Otra situación que se plantea en (Hazen et al., 2002) es la capacidad de responder razonablemente cuando los errores de reconocimiento causan que el módulo de comprensión proporcione sólo un análisis parcial de la intervención del usuario. Por ejemplo, si sólo se ha reconocido la petición de pronóstico de lluvia en cierta fecha y se ha perdido el dato relativo al lugar, el gestor de diálogo decidirá preguntar al usuario por el dato que le falta, la ciudad, y, si no hay más errores, responderá correctamente a la petición del usuario sólo con un turno adicional.

- (Bouwman y Hulstijn, 1998)

En (Bouwman y Hulstijn, 1998), para el sistema PADIS (*Philips Automatic Directory Information System*), y en (Bouwman et al., 1999), para ARISE-NL, se describe la incorporación de las medidas de confianza en los SDS. Las medidas de confianza asignadas a los valores de los atributos, en las fases de reconocimiento y comprensión, permiten que el gestor de diálogo pueda elegir, con mejor criterio, entre las estrategias de confirmación implícita y explícita.

En (Bouwman y Hulstijn, 1998) se concluye que la duración del diálogo y la fiabilidad son dos factores esenciales en la estrategia de diálogo, aunque hay un cierto compromiso que impide la optimización de ambos. El coste del diálogo sería un criterio para primar un objetivo u otro. Así, para información de poco coste, como en PADIS, los usuarios parecen inclinarse por aceptar una menor fiabilidad a cambio de disponer de un servicio más rápido, criterio que se seguirá en el nuevo diseño de la estrategia de diálogo.

- Universidad de Vigo (Pérez-Piñar y García, 2005)

En este reciente trabajo, los investigadores de la Universidad de Vigo exponen un enfoque alternativo en la aplicación de las medidas de confianza consistente en el uso de varios módulos de reconocimiento del habla que, simultáneamente, procesen la señal de entrada del usuario. Se han considerado medidas de confianza de tipo acústico y de tipo lingüístico.

En el sistema presentado en este trabajo, cada módulo ASR utiliza dos modelos de lenguaje: un modelo acústico universal (modelo común para todos los ASR) y un modelo específico, adaptado a un tema (*topic*) determinado (modelo exclusivo para cada ASR). Los *topics* considerados son los siguientes: fechas, nombres propios, números (de teléfono, de tarjeta de crédito, etc) y confirmaciones (frases para aceptar o rechazar información del sistema). De esta manera, cada ASR está especializado en el reconocimiento del habla relativa a una parte de la semántica de la tarea propia del sistema de diálogo. Las salidas generadas por todos los ASR son procesadas por un clasificador que utiliza las medidas de confianza (también facilitadas por los ASR) para clasificar la entrada de usuario en el *topic* más probable.

Los resultados experimentales muestran la viabilidad de esta aproximación. Las frases correspondientes a un *topic* determinado son mejor reconocidas por el ASR específico a ese *topic* y peor reconocidas por los otros ASR (por tanto, el clasificador es capaz de identificar el *topic* correcto y elegir la transcripción correcta). Las mejoras en tasa de reconocimiento son apreciables, especialmente en *topics* difíciles como los nombres y las confirmaciones. Estas mejoras redundarán en un mejor funcionamiento del sistema de diálogo en su conjunto. En particular, la identificación correcta de las confirmaciones es una cuestión muy importante para que en el módulo gestor de diálogo se tomen las decisiones más adecuadas para los diálogos en curso.

- University of Colorado (San-Segundo et al., 2001)

En (San-Segundo et al., 2001b) se estudia el uso de medidas de confianza en los niveles de: a) palabra (donde, para asignar una confianza a las palabras reconocidas, se combinan indi-

cadres acústicos con otros dados por el modelo de lenguaje); b) frase (donde se detectan frases ajenas al dominio, combinando los anteriores indicadores con otros, procedentes del analizador); c) concepto (nivel en el que también se combinan todas las fuentes de información disponibles).

Se informa de una experimentación, realizada sobre un corpus de tamaño apreciable (11.500 frases o *utterances*, en 900 llamadas, realizadas al sistema CU Communicator, de la Universidad de Colorado). La combinación de todos los indicadores considerados, a través de una red neuronal (perceptrón multicapa), permite detectar palabras mal reconocidas, frases fuera de dominio y conceptos incorrectos con tasas de acierto superiores al 50% y sólo con el coste de un 5% de falso rechazo (*false rejection rate*). Además, el uso de las medidas de confianza para elegir las mejores hipótesis en el grafo de palabras, en el módulo de comprensión, permite una mejoría relativa del 14% en *word error rate*.

Posteriormente, en (Hacioglu y Ward, 2002), los investigadores de la Universidad de Colorado han planteado usar un grafo de conceptos, en lugar del de palabras, para establecer las medidas de confianza de las palabras reconocidas. El grafo de conceptos se obtiene a partir del grafo de palabras y de un análisis parcial usando gramáticas (se consideran gramáticas libres de contexto estocásticas, una por concepto, y un modelo de lenguaje de conceptos tipo n-gramas). Como en (San-Segundo et al., 2001b), los resultados que avalan esta propuesta se refieren al sistema CU Communicator.

En (San-Segundo et al., 2001a), en el diseño de la estrategia de confirmaciones del gestor de diálogo, se tienen en cuenta las medidas de confianza. Se establecen 3 umbrales que delimitan 4 niveles de confianza a los que se asocia distintas estrategias de confirmación. Así, para el caso de los atributos origen y destino, en un servicio de información de horarios de trenes, se considera:

- ❑ Si los dos atributos tienen confianza muy alta, entonces se hará una confirmación implícita conjunta;
- ❑ Si uno de los atributos tiene confianza alta (pero no muy alta), entonces se hará una confirmación explícita conjunta;
- ❑ Si uno de los atributos tiene confianza baja, entonces se harán confirmaciones explícitas independientes (en turnos sucesivos);
- ❑ Si uno de los atributos tiene confianza muy baja, entonces se elige la estrategia de rechazo, también independiente, consistente en un mensaje de no entendido seguido de una pregunta del valor del atributo, sin citar ese valor tan inseguro;
- ❑ En el caso de extrema dificultad en la comprensión, se habilita la solicitud de deletreo del nombre de la ciudad.

Además, en el planteamiento de mejora iterativa (dentro del modelo de ciclo de vida) del gestor de diálogo, se establece que cuando el número de veces que el usuario corrija al sistema sea muy alto, se conmute a una estrategia de confirmación explícita, y que cuando éste sea bastante bajo, se tienda al uso de la confirmación implícita.

- Aachen University of Technology (Wessel et al., 1998)

En (Wessel et al., 1998) y (Wessel et al., 1999), se presentan medidas de confianza de carácter estocástico, como alternativa a medidas de otro tipo, como la estabilidad acústica, a las que supera en la evaluación. Se estima la probabilidad a posteriori de una palabra mediante la suma de las probabilidades de todas las hipótesis que contienen esa misma palabra en, aproximadamente, el mismo segmento temporal. Se consideran dos técnicas para el cálculo de las probabilidades: a) por grafos de palabras; b) por listas de mejores hipótesis (*N-best*). Los resultados experimentales, sobre tres corpus diferentes (uno de ellos, el corpus holandés del proyecto ARISE), muestran la superioridad de los grafos de palabras.

En (Macherey et al., 2003a), se describe un mecanismo de detección de errores en los módulos de reconocimiento (ASR) y de comprensión (NLU) y su aplicación para adaptar la estrategia seguida en el módulo gestor de diálogo. En el ASR, se consideran las probabilidades a posteriori de las palabras, calculadas por el método de grafos de palabras (Wessel et al., 1999). En cambio, en el NLU, se calcula la probabilidad a posteriori de los conceptos, dada la secuencia de palabras de entrada, aplicando listas *N-best*.

En el gestor de diálogo, la estrategia de confirmación está supeditada a los errores detectados en las fases previas (a las medidas de confianza facilitadas por los módulos ASR y NLU, como indicadores de posibles errores). El tratamiento es diferente según se consideren las medidas dadas por uno u otro módulo. Para gestionar las confianzas del ASR se consideran dos umbrales que delimitan tres intervalos, asociados a sus correspondientes acciones de diálogo:

- ❑ Si confianza $\in [0, \tau_1[$, entonces confirmación explícita;
- ❑ Si confianza $\in [\tau_1, \tau_2[$, entonces confirmación implícita;
- ❑ Si confianza $\in [\tau_2, 1]$, entonces no se requiere confirmación.

En el caso de las confianzas del NLU sólo se considera un umbral, que determina si se inicia, o no, una confirmación explícita.

En (Macherey et al., 2003a), la experimentación se ha llevado a cabo con el sistema TELDIR, un asistente de directorio telefónico (tarea similar a PADIS). En los resultados se indica la conveniencia de un compromiso entre las medidas *false rejection rate* y *false acceptance rate*, al igual que señalan otros investigadores (Bouwman et al., 1999). Para evaluar el sistema, se analizaron manualmente 35 diálogos, determinando que en un 88% de las sesiones el sistema elegía la acción correcta y éstas concluían satisfactoriamente.

2.3 Gestor del diálogo

En esta sección se revisan algunas aportaciones relevantes en el área de la gestión del diálogo, en particular desde una aproximación estocástica. Se tratan las soluciones dadas a la estrategia de diálogo, los modelos de diálogo, la historia del diálogo y, en general, la representación de la estructura de la tarea. Los trabajos que se comentan son los siguientes:

AT&T (Levin et al., 1999)

Radboud University Nijmegen (Sturm et al., 1999)

CMU (Rudnicky et al., 1999)

University of Colorado (Ward y Pellom, 1999)

LIMSI (Lamel et al., 2000)

MIT (Zue et al., 2000)

Philips (Sougvinier et al., 2000)

Universidad Politécnica de Madrid (Córdoba et al., 2001)

Aachen University of Technology (Macherey et al., 2003)

El aspecto más determinante en el módulo gestor del diálogo es la estrategia de diálogo que se seguirá. Por ello, es habitual asunto de discusión en la literatura, centrándose en el mismo diversos artículos como (Lamel et al., 2000b), (Rosset et al., 1999) y (Sougvinier et al., 2000).

Los modelos de diálogo deben de caracterizar, mediante estados, las situaciones posibles de diálogo en la tarea considerada, y mediante relaciones, conexiones o transiciones, las secuencias de estados que describan cualquier diálogo posible en dicha tarea.

Llamamos historia de un diálogo a cualquier información que se haya obtenido y/o facilitado, por parte del sistema, o por parte del usuario, en cualquiera de los turnos previos y el turno actual de un diálogo determinado.

- LIMSI (Lamel et al., 2000)

En (Lamel et al., 2000b) y (Rosset et al., 1999), los investigadores del LIMSI describen su metodología de diseño y prueba de diferentes estrategias así como exponen la estrategia implementada en su sistema ARISE, una estrategia de iniciativa mixta, en dos niveles, de manera que se permite más libertad al usuario cuando el diálogo se desarrolla correctamente y se restringe su iniciativa, tomándola el sistema, cuando se detectan problemas en la comunicación.

En (Rosset et al., 1999), se distinguen diferentes clases de estrategias, o de principios que las determinan:

- ❑ Estrategias dependientes de aspectos ergonómicos o de usabilidad, como la flexibilidad permitida, las capacidades de negociación y navegación, el tipo de iniciativa, o el contacto con el usuario para mantenerle al corriente del progreso del diálogo.
- ❑ Estrategias dependientes de la metacomunicación, como la identificación de tarea o de cambio de tarea (aspecto estrechamente relacionado con la navegación), y la detección de errores (iniciada por el sistema si se detecta información contradictoria, o por el usuario si cambia de idea o si detecta un error del sistema).
- ❑ Estrategias dependientes del conocimiento de la tarea, que determinan la interdependencia entre las restricciones. Por ejemplo, para la subtarea de reservas es condición necesaria completar antes la subtarea de selección de un tren específico (puesto que el precio depende del horario y del tipo de tren). Siguiendo con el ejemplo, ante la tarea de reserva, el sistema puede adoptar diferentes estrategias: a) preguntar siempre si el usuario desea hacer una reserva; b) suponer, de entrada, que ésa será la intención del usuario.

En todo diálogo se distinguen varias fases (ver (Lamel et al., 2000b) y (Rosset et al., 1999)):

- ❑ Adquisición: el sistema obtiene la información necesaria para completar la tarea.
- ❑ Negociación: cuando el usuario introduce cambios en función de la información que le devuelve el sistema (como, por ejemplo, la relajación de las restricciones de búsqueda después de que el sistema informe de que no hay información en la BD que satisfaga la petición precedente).
- ❑ Navegación: cuando el usuario cambia de una tarea a otra, o cuando pregunta por la funcionalidad del sistema o los tipos de información disponible.
- ❑ Post-aceptación (*post-acceptance*): cuando el usuario acepta la solución propuesta, el diálogo puede pasar a abordar otra tarea, sea por petición del usuario o por sugerencia del sistema. Si esta posibilidad se rechaza, se alcanza el cierre del diálogo.
- ❑ Metacomunicación: fase relativa a la detección y tratamiento de los errores.

El gestor de diálogo usa y mantiene varias fuentes de conocimiento que determinan el estado del diálogo: la historia de pares intercambiados entre sistema y usuario, la historia completa del diálogo y un modelo de la tarea.

La estrategia de recuperación de errores es diferente según la fase del diálogo en la que se producen: a) en la fase de adquisición, se puede verificar cualquier información contradictoria; b) en la fase de negociación, sólo se permiten cambios en la hora y restricciones adicionales (pues se supone que ya se han confirmado los atributos básicos: origen, destino y fecha del viaje). El sistema iniciará subdiálogos restringidos, orientados a obtener un dato específico, cuando detecte dificultad en obtener alguno de los atributos básicos o cuando exista una incoherencia entre lo almacenado por el sistema y el nuevo dato recibido. En cualquier caso, la corrección de errores es una cuestión compleja, dado que éstos también puede presentarse durante los mismos subdiálogos iniciados para su tratamiento.

Se puede decir que (Lamel et al., 2000b) es una recapitulación de aspectos generales de las investigaciones realizadas en el LIMSÍ, ya expuestos en (Lamel et al., 2000a) y (Rosset et al., 1999).

- Radboud University Nijmegen (Sturm et al., 1999)

En (Sturm et al., 1999a) y (Sturm et al., 1999b), investigadores responsables del ARISE-NL discuten varios aspectos relativos a la estrategia de diálogo a la luz de las experiencias con los primeros prototipos del sistema. Las conclusiones principales de su estudio son:

- ❑ Una estrategia de confirmaciones condicionada a las medidas de confianza. El uso exclusivo de una estrategia de confirmación implícita resulta problemático, si hay errores de reconocimiento, pues el usuario no acierta a corregir al sistema cuando, al mismo tiempo, se le solicita nueva información. Por otra parte, usar sólo una estrategia de confirmación explícita (ARISE-NL '98) facilita la corrección de errores y no incrementa la duración del diálogo, en segundos, porque permite respuestas muy cortas. Sin embargo, aumenta el número de turnos y los usuarios juzgan que la interacción es tediosa. Estas observaciones han llevado (ARISE-NL '99) a aplicar una combinación de confirmaciones, implícitas y explícitas, en función de la confianza en el reconocimiento de los datos. Así, si el dato tiene una confianza alta, se confirmará implícitamente; sino, se usa una confirmación explícita.
- ❑ Iniciativa mixta, superior a una interacción dirigida por el sistema. La iniciativa mixta permite el desarrollo de diálogos más naturales. Los usuarios expertos pueden reducir la duración del diálogo, proporcionando antes los datos de la consulta. Los usuarios noveles pueden dejarse guiar, respondiendo a todas las preguntas del sistema, pero también éstos usan su iniciativa en casos como el rechazo a un dato incorrecto que, habitualmente, se acompaña con la corrección o aporte espontáneo del dato válido.
- ❑ El manejo de excepciones (*exceptions handling* o *zooming*). Conforme a la estrategia en dos niveles, característica de ARISE, cuando no se consigue (por error de reconocimiento o por falta de iniciativa propia) que el usuario suministre la información necesaria, el sistema conmuta a la estrategia dirigida, proporcionando consejos a cerca de las opciones disponibles.
- ❑ Navegación flexible, como aproximación a una completa capacidad de negociación (no disponible en los actuales SDS). El sistema ARISE responde con la información relativa a un único viaje, el que mejor se ajuste a la consulta del usuario. A continuación, se abre un subdiálogo de navegación donde el usuario podrá preguntar por trenes anteriores o posteriores, conexiones con algunos cambios, viajes de vuelta, otras conexiones, etc. Si el usuario acepta esta navegación (porque no le interesa la primera respuesta del sistema) pero no ejerce su iniciativa, el sistema aplica *zooming* y el usuario entrará en un menú. Pese a las mejoras en la parte de negociación, ésta se juzga aún demasiado rígida.
- ❑ Defectos en el diseño de los escenarios o en su interpretación. Si bien el uso de escenarios tiene la ventaja de controlar el uso del sistema y simplificar su evaluación, se aprecia que es muy difícil presentar los escenarios de modo que no sean ambi-

guos ni tampoco sugieran expresiones concretas a los usuarios. Por otra parte, en los escenarios complejos, con varias tareas, se observó que bastantes usuarios no los completaban porque no intentaban llevar a cabo el subdiálogo de navegación (las razones serían varias: no comprendieron que eran tareas a cumplir, tenían más tendencia a aceptar información incorrecta porque realmente no la necesitaban, o no se daban cuenta de que obtenían información incorrecta).

En cuanto a modelos de diálogo, en (Baggia et al., 1999) se discuten distintas técnicas para obtener modelos de lenguaje adecuados para los SDS, teniendo en cuenta aspectos como el tamaño reducido de los corpus de entrenamiento o como las restricciones disponibles en un sistema de diálogo. Se plantean las siguientes alternativas:

- ❑ Modelos de segmentos de conceptos (*conceptual segment modelling*). Permiten una comprensión automática robusta, siendo especialmente eficientes para tratar problemas de ambigüedad en frases y palabras en francés.
- ❑ Modelos dependientes de estados de diálogo (*dialogue-state dependent language models*). Presentan la dificultad del elevado número de posibles estados de diálogo, que llevaría a crear muchos modelos, y algunos correspondientes a estados apenas presentes en el corpus. Además, existen grupos de estados de diálogo que conducen a respuestas del usuario muy parecidas. Se comentan diversas soluciones: interpolación, clasificación automática, uso de conocimiento adicional...
- ❑ Modelos iniciales sintéticos (*bootstrap language models*). Se obtienen cuando se dispone de muy pocos datos de entrenamiento, al aplicar métodos Monte-Carlo para incrementar automáticamente la cantidad de datos. A partir del corpus artificial así generado, pueden entrenarse modelos de n-gramas.

- MIT (Zue et al., 2000)

El gestor de diálogo (*turn manager*) de JUPITER (Zue et al., 2000) no ejerce el decisivo control que este módulo tiene en otros sistemas. Parece que, en la arquitectura GALAXY II, muchas de las funciones habituales de este módulo han sido transferidas al *hub* programable a través del que se conectan todos los módulos.

Aún así, al *turn manager* se le asigna el cometido de interpretar la petición del usuario y preparar la respuesta del sistema. Sin embargo, la necesaria comunicación con los otros módulos (la aplicación *back-end* de acceso a la BD, el analizador TINA, el generador de respuestas GENESIS...) se encuentra regulada por el *hub*.

Para dotarlo de cierta independencia respecto a la tarea, el comportamiento de este módulo se ajusta por medio de una tabla de control de diálogo, externa a su código. En dicha tabla se especifican: a) los conjuntos de condiciones necesarias para llevar a cabo ciertas acciones; b) las secuencias de operaciones que se activarán si se cumple un conjunto determinado de condiciones. Esta tabla, por tanto, recopila un conjunto de reglas que gobiernan la conducta del sistema.

- CMU (Rudnicky et al., 1999)

En (Rudnicky y Xu, 1999a) y (Rudnicky et al., 1999b), se describe el gestor de diálogo integrado en el Carnegie Mellon Communicator. Se ha diseñado como un módulo independiente de la tarea y cuyo comportamiento se fija mediante un *script* dependiente de la tarea, además de por la acción complementaria de tres agentes de dominio (agente de información de viajes, agente de interpretación de fechas y horas, y agente de modelo de usuario).

El análisis de diálogos de planificación de viajes, mantenidos por personas, ha permitido implementar un conjunto de guiones (*scripts*) que capturan las actividades convencionales asociadas a esa tarea compleja. Estas actividades, o subtareas específicas, constituyen las planificaciones (*schema*) que se agrupan para formar los *scripts*, estructuras que proporcionan al sistema los modos de interaccionar con el usuario. Además de los *scripts*, el sistema usa una estructura de datos objetivo (*target*) donde se almacena el producto (*product*) de la interacción.

Se trata, por tanto, de un gestor de diálogo basado en *scripts*. Un *script* determina una secuencia explícita de conceptos propios de la tarea. Cada concepto (*topic*) se concreta en una tarea de rellenar un formulario (*form-filling*), abordada con iniciativa mixta, pudiendo el usuario seguir el orden que desee para suministrar los valores. Por parte del sistema, existe una estrategia según la cual se preguntará al usuario por las posiciones vacías del formulario (*slots*) en función de su relevancia para determinar o acotar la información solicitada.

La estructura basada en *schema* puede usarse tanto para obtener estrategias por defecto que aborden diferentes objetivos parciales como para establecer los detalles de la interacción en situaciones específicas.

En cuanto a la estrategia, se consideraron dos políticas. Una política es rígida, pues se obliga al usuario a responder preguntas en un orden fijo (iniciativa del sistema); la otra, libre, pues permite al usuario aportar información (iniciativa mixta). La ocurrencia de entradas incorrectas dictaría cuándo se conmutaría de la iniciativa mixta a la otra más rígida.

En (Rudnicky y Xu, 1999a) se plantean, también, las limitaciones de la aproximación basada en *scripts*. Estos están demasiado relacionados con la estructura de datos a rellenar (*product*), estructura que resulta ser demasiado estática, de tipo formulario. En la tarea de planificación de viajes, conviene disponer de una estructura dinámica, que cambie a lo largo de la sesión, y que facilite su navegación. Por todo ello, en (Rudnicky y Xu, 1999a), se introducen nuevas estructuras de datos:

- Un producto (*product*) dinámico, representado por un árbol. Se permite modificar su estructura, en el curso de una sesión, añadiendo ramas que representen nuevos itinerarios que solicite el usuario. Para ello, se define una librería de subárboles y mecanismos para vincularlos a la estructura del producto. A cada nodo del árbol se asocia un manejador (*handler*), que consta de: a) un conjunto de receptores asociados a las redes de entrada; b) una transformación a aplicar para obtener un valor; c) una especificación de lo que el sistema puede decir sobre la información que se maneja en el nodo. Un *handler* equivale a un *schema*, y gestiona un subdiálogo acerca de una información específica.
- Una agenda, que viene a ser una pila (*stack*) generalizada, sustituye al *script* fijo. La agenda es una lista ordenada de conceptos (*topics*) que especifica la estrategia

completa para llevar a cabo la tarea. Cuando se recibe una nueva entrada del usuario, la agenda explora el árbol producto de izquierda a derecha y en profundidad, invoca, según este orden, a los manejadores, y éstos tratan de interpretar la entrada del usuario. La agenda permite adaptar dinámicamente el producto (conforme a la iniciativa del usuario en una sesión dada). Así, cuando un manejador se convierte en foco de atención, éste es promocionado a la posición superior de la agenda y, además, para mantener los conceptos en su contexto, se aplica una operación similar a la rama que contiene al nodo (*sub-tree promotion*). La flexibilidad es completa, pues tanto el sistema como el usuario pueden reordenar los elementos en la agenda.

En cuanto a modelos de diálogo, en (Rudnicky y Xu, 1999a) se consideran dos aproximaciones:

- ❑ Sistemas basados en grafos. Donde se enumeran todos los posibles estados de diálogo y las transiciones permitidas, según la ocurrencia de sucesos específicos. Los grafos suelen ser árboles y presentan limitaciones en su uso, como dificultad para conmutar a otro concepto, codificado en un subárbol diferente, o dificultad para navegar en un árbol dado.
- ❑ Sistemas basados en frames. La gestión del diálogo consiste en rellenar formularios. Hay que supervisar que el formulario se complete, fijar los elementos dados por el usuario y preguntar al usuario por los que no haya facilitado. Son más flexibles en la obtención de datos, pues no se requiere seguir un orden particular (a diferencia de lo que ocurre en el recorrido de los árboles), pero menos en cuanto que no permiten modificar la estructura de los formularios.

Para la tarea de planificar viajes, en el CMU Communicator, no resulta adecuado el procedimiento de completar un formulario puesto que no se conoce de antemano la estructura exacta del viaje, estructura posiblemente compleja si consta de varios itinerarios. Por ello, la tarea se modela mediante árboles, aunque no se abandona el concepto de formulario. Un itinerario complejo será una estructura jerárquica de formularios.

En cuanto a la historia del diálogo, en (Rudnicky et al., 1999b), el conocimiento de la tarea de planificación de viajes se formaliza en dos estructuras de datos: a) un producto (*product*) que guarda el resultado de la interacción; b) un conjunto de *schema*, que describe el modo de acceso a los elementos del producto. Dada la tarea, las entidades más importantes del producto son los itinerarios.

Un itinerario es una estructura de datos jerárquica o arborescente. El árbol del itinerario es construido interactivamente, con las restricciones impuestas por el usuario y con el conocimiento del dominio de que dispone el sistema. Construir un itinerario requiere dos actividades: a) definir la estructura del árbol; b) asignar valores a esa estructura con datos específicos de la tarea. Los constructores de *schema* se usan para construir y editar la estructura del producto.

- University of Colorado (Ward y Pellom, 1999)

En (Ward y Pellom, 1999), se presenta el sistema CU Communicator, desarrollado en el Center for Spoken Language Understanding, de la Universidad de Colorado. La arquitectura del sistema es similar a la de Galaxy, con un módulo *hub* que conecta todos los servidores y retransmite los mensajes que intercambian. Usan los mismos módulos de reconocimiento y de comprensión que en el sistema de la CMU: Sphinx-II y una versión modificada de Phoenix, respectivamente.

En su gestor de diálogo se ha implementado una estrategia flexible, que no obedece a un guión o conjunto de reglas, sino que está dirigida por sucesos que afecten al cambio del contexto del diálogo (*event driven dialog manager*). Entre las funciones de este módulo se citan:

- ❑ Integrar la interpretación de la entrada (frames recibidos) con el contexto del diálogo;
- ❑ Resolver ambigüedades de interpretación, estimar la confianza de la información obtenida;
- ❑ Solicitar al usuario aclaraciones sobre la interpretación o el contexto, pedir al usuario la información que falte;
- ❑ Construir la consultas a la base de datos (*SQL queries*);
- ❑ Enviar información al módulo de generación de lenguaje natural para su presentación al usuario.

En este gestor de diálogo no se consideran modos separados de iniciativa del usuario o iniciativa del sistema. Cuando el sistema dispone de suficiente información para atender la consulta, lo hace, y si no, pregunta por lo que necesita. El usuario, sin embargo, puede responder lo que quiera, y el sistema, tras analizar la posible nueva petición, fijaría su objetivo en el frame resultante.

La estrategia *event driven* seguida consiste en que, una vez integrada la interpretación de los frames de usuario con el contexto del diálogo, el gestor de diálogo examina este contexto para decidir la acción a llevar a cabo. Sus alternativas, listadas en orden de prioridad, son las siguientes:

- ❑ Resolución de ambigüedades (*Clarify if necessary*). Interacción con el usuario para obtener una interpretación única y fiable.
- ❑ Cierre si la consulta ha terminado (*Sign off if all done*).
- ❑ Acceder a la BD e informar al usuario (*Retrieve data and present to user*). Obtener los resultados de la consulta a la BD y presentar al usuario la información de la mejor tupla.
- ❑ Petición de información (*Prompt user for required information*). Solicitar los datos que falten para construir la consulta a la BD. Se establece una escala de prioridades en la petición de datos para rellenar los *slots* vacíos.

El acceso a la información de la BD a través de Internet plantea riesgos de retrasos, a veces prolongados, en el curso del diálogo. Por ello, el gestor de diálogo comprueba que los registros solicitados no se encuentren ya en su memoria local, antes de enviar la petición a través de la red. También se fija un límite de espera (30 segundos) a la obtención de la página *html* de resultados. Si se excede, el gestor de diálogo avisa al usuario de que la causa de la demora está en el acceso a Internet.

Además, se ha procurado que el gestor de diálogo sea bastante independiente de la tarea, mediante una representación declarativa. Así, la mayoría de la información específica del dominio se encuentra en ficheros externos al código C del módulo.

En cuanto al uso de la historia del diálogo, en (Ward y Pellom, 1999), la estrategia del gestor de diálogo se encuentra determinada por dicha historia, aquí denominada contexto actual. El contexto es un conjunto de estructuras de C, tratándose de una representación conceptual basada en formularios.

Cada formulario consta de un conjunto de campos, y cada campo de un conjunto de punteros: 1º, al nombre del campo; 2º, a su valor; 3º, al análisis de comprensión (frame) que se corresponde con el campo; y otros punteros, a plantillas para generar salidas que usen el valor del campo, tanto en lenguajes formales (por ej., SQL, para consultar la BD) como naturales, para el generador de respuestas. Este mecanismo permite una fácil ampliación para aplicaciones multilingües.

- *Aachen University of Technology (Macherey et al., 2003)*

En (Macherey y Ney, 2003b), los investigadores de la Universidad de Aachen también orientan su trabajo hacia el desarrollo de una arquitectura de gestión de diálogo independiente del dominio y que se basa en el uso de estructuras de datos arborescentes.

Mediante el árbol se representa el conocimiento asociado a una tarea específica (en el caso del sistema TELDIR, un asistente de directorio telefónico). De este modo, las acciones del gestor de diálogo se pueden definir como operaciones sobre árboles, con cierta independencia del dominio de la tarea.

A partir del árbol original, durante cada sesión de diálogo, se generan árboles específicos (*instance trees*), donde se almacenan los pares de conceptos y atributos dados en la entrada del usuario. En cada momento en que el sistema tenga que actuar, se encontrará en una de las siguientes situaciones:

- ❑ Existe un solo camino, desde la raíz a alguna hoja del árbol, tal que todos los nodos del camino tienen asignados todos los pares concepto/atributo necesarios. La acción del sistema será responder a la petición del usuario.
- ❑ Existe más de un camino en el árbol que cumpla esas mismas condiciones. Se entiende entonces que la entrada del usuario contiene alguna ambigüedad. La acción del sistema será solicitar al usuario que restrinja su petición.
- ❑ No existe ningún camino que cumpla esas condiciones. Se entiende que existen nodos aún vacíos porque el usuario no ha aportado bastante información. La acción del sistema será pedir la información necesaria para rellenar los nodos que falten.

Es habitual que se disponga de varios caminos válidos para continuar un diálogo. La decisión sobre la siguiente acción de diálogo se toma optimizando una función de coste. Esta función proporciona un valor para todos los nodos del árbol. En (Macherey y Ney, 2003b), se describen los indicadores tenidos en cuenta en el cálculo de la función de coste. Son los siguientes:

- ❑ Medidas de confianza de las palabras. Útiles para evitar turnos superfluos de confirmación.
- ❑ Importancia y asignación (*importance and filling degree*) de los conceptos. Existe una escala de relevancia de los conceptos y atributos, que tiene en cuenta también sus dependencias.
- ❑ Grado de ambigüedad. Para cada nodo se puede señalar si contiene información ambigua.
- ❑ Información contradictoria. También para cada nodo se indica si la contiene, pues ésta lleva a ejecutar una consulta vacía sobre la BD.
- ❑ Número de resultados SQL. Es el número de tuplas que devuelve la consulta. Si es muy grande, el usuario debe restringir su petición. Si es cero, la petición no está soportada o deben relajarse algunas de sus condiciones.
- ❑ Verificación de la información. También para cada nodo se indica si la información que contiene ha sido confirmada.

A partir de los costes de todos los nodos en un particular *instance tree*, se puede calcular el coste de cada camino en el árbol. Y, por tanto, decidir el mejor camino y la siguiente acción de diálogo. Así, si el mejor camino:

- ❑ Contiene información con confianza menor a cierto umbral, la acción será una confirmación explícita.
- ❑ Contiene información ambigua, la acción será una pregunta que permita resolverla.
- ❑ No contiene la información necesaria (hay nodos a completar), la acción será una pregunta sobre esa información.
- ❑ Si el camino está completo y tiene un número moderado de respuestas SQL, la acción será responder a la consulta del usuario.

Se aprecia que la estructura de los árboles de decisión condiciona en exceso la estrategia del diálogo. Por ello, en (Macherey et al., 2003a), se anuncia la intención de sustituir la aproximación basada en árboles de decisión por la aproximación basada en máxima entropía. Es decir, un tratamiento más probabilístico de la gestión del diálogo (en la línea de lo que ya realizan estos investigadores en el campo de la comprensión del lenguaje natural).

En lo relativo a la historia del diálogo, en (Macherey et al., 2003a) y (Macherey y Ney, 2003b), también se representa el conocimiento acerca de la tarea mediante árboles. Para una tarea determinada, se diseña un árbol genérico, árbol de conocimiento de su dominio. En este árbol, los nodos representan conceptos y los arcos representan relaciones entre los conceptos. Durante una sesión de diálogo, el módulo gestor del diálogo recibirá secuencias de conceptos (resultado de la interpretación hecha por el módulo NLU) e incorporará cada

concepto en todos los nodos del árbol que lo incluyan. Así se construye un árbol instanciado (*instance tree*) que refleja la historia del diálogo.

- Universidad Politécnica de Madrid (Córdoba et al., 2001)

En (Córdoba et al., 2001), el Grupo de Tecnología del Habla, de la Universidad Politécnica de Madrid, presenta los resultados obtenidos en el proyecto IDAS (*Interactive Directory Assistance Service*). Por la naturaleza del proyecto, se destaca el módulo de reconocimiento del habla. De todos modos, es interesante observar las características que se recomiendan para el módulo gestor de diálogo.

Dadas las serias dificultades para el ASR (ruido telefónico, gran tamaño del vocabulario, alto grado de similitud entre muchos apellidos en español), y la imposibilidad de disponer de un ASR perfecto, el gestor de diálogo tiene que habilitar soluciones: solicitudes de confirmación al usuario y peticiones de deletreo de nombres. Se ha elegido una estrategia de diálogo dirigida por el sistema, por considerarla la más adecuada para incrementar el éxito de los diálogos.

En (San-Segundo et al., 2001a) se describe la propuesta de este grupo, relativa a la metodología de diseño de los gestores de diálogo. Se trata de una metodología similar al modelo de ciclo de vida, planteada en el proyecto DISC, y que se organiza en 5 pasos:

- ❑ Análisis de la base de datos. Se describe, mediante un diagrama entidad-relación (E-R), la información contenida en la BD, objeto del servicio ofrecido por el SDS. Las entidades estarán asociadas a los objetivos que ha de lograr el sistema. Los campos clave serán los elementos de información que el usuario deberá aportar obligatoriamente y que, por tanto, definirán las interacciones del diálogo.
- ❑ Diseño por intuición (o *brain-storming*). Fase en la que se determinan los objetivos del servicio ofrecido (entidades cuyo valor se consultará) así como las secuencias de acciones y los elementos de información necesarios para alcanzarlos.
- ❑ Diseño por observación. Donde se analizan diálogos usuario-operador humano en una tarea o servicio igual o parecido. El objetivo es depurar las alternativas planteadas en el diseño por intuición, mediante el análisis, en estos diálogos reales, de: a) los objetivos (frecuencia de petición, secuencias de consulta, información dada por el operador para satisfacerlo); b) los elementos de información (secuencias de consulta, clasificación en obligatorios u opcionales, en simples o complejos, modos de especificación y ordenamiento); c) la negociación (información de ayuda a la toma de decisión del usuario, número de alternativas simultáneas que el operador plantea).
- ❑ Diseño por simulación, usando la técnica del mago de Oz. En esta fase se busca mejorar el flujo de diálogo. Para ello, se recopilan medidas acerca de los experimentos realizados: anotaciones del sistema y respuestas de los usuarios a cuestionarios. Esto permite evaluar las decisiones de diseño de las fases previas.
- ❑ Diseño por mejora iterativa. Mediante el estudio del funcionamiento del prototipo de SDS automático, se pueden plantear y refinar: a) estrategias de confirmación (implícita, explícita, rechazo...); b) mecanismos de recuperación de errores (co-

mandos especiales para corrección puntual o para reinicio de la sesión); c) adaptación al usuario (se establecen 4 niveles de habilidad de usuario, y se puede conmutar de nivel en función del número de errores, o de confirmaciones positivas, que se produzcan).

- AT&T (Levin et al., 1999)

Dentro de la aproximación estocástica, en (Levin et al., 1999) y (Levin et al., 2000a), se presenta el algoritmo del gestor de diálogo, formalizado como un proceso de decisión secuencial (ver la figura 1 de (Levin et al., 1999)):

```

St = SI
while St <> SF {
    At = NextAction(St)
    invoke At
    Ot = environment response to At
    St+1 = NextState(St, At, Ot)
    t = t+1
}

```

En (Levin et al., 1998) y (Levin et al., 2000a), se describen las sucesivas estrategias obtenidas en un proceso de aprendizaje automático de las mismas, dirigido por la minimización de una función objetivo que mide el coste del diálogo. A continuación, se enumeran estos hitos en el aprendizaje de la estrategia y la justificación de que se alcancen:

- ❑ Estrategia de cierre inmediato (*closing*). Esta estrategia, aprendida tras 20 diálogos, tiene un coste muy alto, debido a la obvia insatisfacción del usuario. Sin embargo, tiene menos coste que una estrategia aleatoria (punto de partida del aprendizaje).
- ❑ Estrategia con apertura de cortesía (*greeting*), a la espera de la iniciativa del usuario. Aprendida después de 12.000 diálogos, porque aumenta el número de restricciones (datos facilitados por el usuario) en el primer turno y, en consecuencia, reduce la duración del diálogo.
- ❑ Estrategia con estado opcional de restricción (*constrain*). Aprendida tras 90.000 diálogos, porque formular preguntas que restrinjan el rango de la consulta supone que se reduzca el coste de recuperación de información de la base de datos.
- ❑ Estrategia con estado opcional de relajación (*relax*). Aprendida tras 700.000 diálogos, al comprobar que ampliar el rango de la consulta evita las situaciones anómalas de obtener conjuntos vacíos de información de la base de datos.

En (Levin et al., 2000b), se describe la estrategia implementada en DARPA Communicator. Se establece un compromiso entre la iniciativa permitida al usuario y el rendimiento del ASR. De entrada, el sistema permite el máximo de iniciativa al usuario. Sin embargo, cuando se detectan problemas (como la petición reiterada de un mismo atributo), el sistema gradualmente reduce esa iniciativa, aplicando modelos de lenguaje más restrictivos que se reflejan en preguntas al usuario cada vez más específicas. Si, tras varios intentos, el problema persiste, se conmuta al modo de iniciativa estricta del sistema, donde cada atributo

deberá ser explícitamente confirmado. Se trata, por tanto, de una estrategia que coincide claramente con la estrategia de dos niveles del LIMSÍ.

La estrategia, o flujo funcional del diálogo, en la tarea de planificación de viajes implementada en DARPA Communicator es la siguiente:

- ❑ Identificación de usuario (*sign in*). Fase inicial, opcional, que permite al usuario activar su perfil. Existe un usuario invitado, sin perfil particular.
- ❑ Planificación de vuelo (*flight planning*). Constituye la parte principal del diálogo y, para cada itinerario, se estructura en dos fases:
 - 1ª) Recopilación (*information gathering*). El sistema solicita todos los datos obligatorios para construir la consulta a la BD. El usuario puede tomar la iniciativa y proporcionar varios datos a la vez, incluyendo datos opcionales que el sistema también añadirá a la consulta a la BD.
 - 2ª) Presentación y negociación (*flight presentation and negotiation*). Tras recopilar la información necesaria y lanzar la consulta a la BD, el sistema presenta los resultados. Si la consulta no proporciona resultados, el sistema toma la iniciativa de relajar alguna condición relativa a atributos opcionales e informa al usuario de ello. Si la consulta proporciona muchos resultados, éstos, en principio, se ordenan por precio. El usuario, entonces, tiene las opciones de: a) aceptar el resultado presentado; b) recorrer la lista de resultados; c) filtrar la lista, proporcionando restricciones suplementarias.
- ❑ Planificación de servicios en destino (*ground arrangements*). Constituye una fase opcional del diálogo donde el usuario puede reservar hoteles o coches. Se estructura en dos fases como la planificación del vuelo.
- ❑ Presentación de resumen de itinerario (*itinerary summary presentation*). Opcional, a petición del usuario.
- ❑ Encuesta y cierre (*user satisfaction polling and closing remarks*).

En resumen, la estrategia de diálogo en (Levin et al., 2000b) consiste en desglosar una tarea compleja en subtareas. Cada subtarea tiene un objetivo específico y se desarrolla un subdiálogo que busca cumplirlo. Con independencia de su objetivo, los subdiálogos tienen un esquema común: primero, recopilación de datos; segundo, presentación de resultados y posible negociación.

En cuanto al uso de la historia del diálogo, en (Pieraccini et al., 1997), en la arquitectura AMICA, el curso de diálogo depende de la historia (*dialogue state*), que guarda la información disponible desde el inicio del diálogo en una estructura de pares {nombre, valor}, y el estado (*control state*), que identifica una situación en el diagrama de flujo que regula el diálogo.

En (Levin et al., 2000a), la representación del estado del sistema se estructura en tres plantillas que reflejan la historia del diálogo:

- ❑ Plantilla de usuario (*user template*). Representa el significado de la petición del usuario, interpretada en su contexto. Consta de una lista de pares {nombre, valor} de todos los atributos.
- ❑ Plantilla de datos (*data template*). Describe los datos extraídos de la base de datos, conforme a una consulta basada en la plantilla de usuario. Consta de un cierto número de tuplas de datos.
- ❑ Plantilla del sistema (*system template*). Representa la historia de acciones del sistema durante el diálogo.

- *Philips (Sougvinier et al., 2000)*

En (Sougvinier et al., 2000), se discuten estrategias que compatibilicen la robustez del sistema con un adecuado nivel de naturalidad en el curso del diálogo. La estrategia considerada se basa en dos principios: a) considerar las distintas fuentes de información tan pronto como sea posible; b) mantener alternativas, retrasando la elección de una única opción tanto como sea posible. Consecuencia del primer principio es una arquitectura del sistema donde información procedente del gestor de diálogo o del sistema gestor de la BD puede ser usada para valorar las hipótesis generadas en los módulos de reconocimiento y comprensión.

En los sistemas desarrollados en Philips, el gestor de diálogo sigue una estrategia consistente en completar los datos necesarios para la consulta (*slot-filling strategy*). Al tratarse de sistemas con iniciativa mixta, el usuario puede proporcionar valores para varios *slots*, con independencia de que éstos figurasen en el precedente *prompt* del sistema. La tarea del gestor de diálogo es rellenar los *slots* necesarios para poder satisfacer la petición del usuario, procurando que el diálogo sea lo más breve posible.

Una cuestión relevante que se plantea es el uso de los distintos tipos de confirmación. Lo más seguro, pero también más tedioso y menos natural, es la confirmación explícita de cada uno de los valores asignados a los *slots*. La confirmación implícita, seguida de ausencia de corrección por parte del usuario, resulta una técnica más refinada, aunque puede no ser recomendable para algunas acciones críticas (como pasar una llamada, en un sistema directorio como PADIS).

Además, se contempla cómo las medidas de confianza sobre las unidades semánticas pueden afectar a la estrategia del diálogo.

Con la intención de facilitar la migración a otras aplicaciones, se ha desarrollado HDDL, un lenguaje de descripción de diálogos, que permite especificar, de modo declarativo, las definiciones de los *slots*, las preguntas y las estrategias de confirmación.

En este trabajo, la historia del diálogo, o creencias del sistema (*system belief*), consiste en una o más hipótesis sobre los valores de los *slots* y su estado de confirmación. El lenguaje HDDL (definido para facilitar la adaptación del gestor de diálogo a diferentes dominios) proporciona comandos para gestionar los *slots*, en particular para determinar cuáles *slots* deben ser borrados y cuáles mantener sus valores en cada momento de una sesión de diálogo.

2.4 Generación de respuestas

Los procedimientos más habituales para construir las respuestas en lenguaje natural que el sistema proporcionará a sus usuarios son el uso de reglas (*linguistic rule-based approach*) y el uso de plantillas predefinidas (*template-based approach*). Ambas, sin embargo, presentan limitaciones al aplicarse a un SDS. Los sistemas basados en reglas, usualmente genéricas, son difíciles de adaptar a aplicaciones que aborden tareas específicas. El coste de escribir y mantener las reglas de la gramática es elevado. Los sistemas basados en plantillas requieren un esfuerzo algo menor. Aún así, y como la calidad del sistema depende directamente de la talla del conjunto de plantillas, el coste de mantenimiento también es alto si se quiere proporcionar una salida de calidad razonable. Recientemente, se pueden encontrar trabajos que abordan la generación de respuestas mediante métodos estadísticos (*corpus-based approach*) y tratan de superar las anteriores limitaciones.

El sistema GALAXY del MIT se basa en el uso de un conjunto de reglas, que incluyen los *prompts* de salida del sistema para los siguientes turnos. El sistema de Philips (Aust et al., 1995) usa un conjunto predefinido de plantillas de subdiálogo, que serán llamadas o importadas según se necesiten.

- MIT (Zue et al., 2000)

En (Zue et al., 2000), se describe sucintamente GENESIS, el módulo de generación de lenguaje desarrollado en el MIT, en su integración en el sistema JUPITER. La entrada a GENESIS siempre es un frame semántico, pero éste puede proceder de: a) TINA, el módulo de comprensión; b) el gestor de diálogo (*turn manager*) de JUPITER; c) ambos módulos.

Para traducir los frames recibidos, GENESIS dispone de:

- ❑ Un diccionario (*lexicon*) que especifica la forma superficial, incluyendo inflexiones, asociada a la entrada semántica.
- ❑ Un conjunto de plantillas de mensajes, que se usan recursivamente para construir frases que describan los temas, predicados y cláusulas del frame.
- ❑ Un conjunto de reglas de reescritura, usadas para aplicar ciertas contracciones y restricciones fonético-sintácticas.

GENESIS traduce los frames a frases en lenguaje natural (inglés), pero también a SQL. Así, cuando el *turn manager* de JUPITER está preparado para acceder a la BD, envía su petición a GENESIS, y éste construye una consulta SQL de sintaxis correcta.

- CMU (Oh y Rudnicky, 2000)

En (Rudnicky et al., 1999b), las intervenciones del sistema se generan usando plantillas basadas en locuciones (*phrases*) y oraciones (*sentences*). Se incorpora, además, un mecanismo de control de la historia del diálogo que posibilita modificar la forma derivada de las

plantillas, filtrando algunos atributos que aparezcan en esa salida. Este mecanismo se usa para controlar la generación de confirmaciones implícitas y para gestionar el uso de información nueva y antigua, según los conceptos (*topic*).

En (Rudnicky y Xu, 1999a), dada la introducción de nuevas estructuras de datos (*agenda, handlers, product trees*) en el gestor de diálogo, se introducen algunos cambios en el módulo de generación de lenguaje. Así, asociados a cada manejador (*handler*) hay un conjunto de llamadas a este módulo, que corresponden a condiciones fijadas dentro del manejador y que especifican un acto de diálogo de salida y un conjunto relevante de conceptos. Además, se informa de que se ha usado un generador estocástico, entrenado con instancias de intervenciones reales de los usuarios.

La aproximación estocástica se encuentra más ampliamente tratada en (Oh y Rudnicky, 2000) y (Oh y Rudnicky, 2002). Los problemas a tratar, según (Oh y Rudnicky, 2000), son dos:

- ❑ Planificación del contenido (*content planning*). Consiste en decidir qué atributos deben incluirse en la intervención (*utterance*) del sistema. Conforme el diálogo progresa, el número de atributos aumenta: incluirlos todos conduciría a frases demasiado extensas; suprimir todos los atributos opcionales, podría confundir al usuario. Se necesita un algoritmo que determine qué debe repetirse y cuándo.
- ❑ Construcción de las frases (*surface realization*). Las intervenciones simples pueden generarse eficientemente con plantillas. Para las intervenciones más complejas se requiere aplicar reglas de una gramática. El objetivo será sustituir la gramática por un modelo de lenguaje estadístico que pueda generar las mismas intervenciones.

Según se expone en (Oh y Rudnicky, 2000), como punto de partida, se definieron dos conjuntos de etiquetas correspondientes a clases de intervenciones y a clases de palabras. A continuación, se etiquetaron dos corpus en el dominio de la planificación de viajes (CMU corpus y SRI corpus). A partir de éstos, se construyeron modelos de lenguajes de n-gramas.

Para la planificación del contenido, se construyeron dos modelos:

- ❑ Modelo de número de atributos. Predice el número de atributos en la intervención del sistema, n_k , dada la clase de intervención, c_k . Esto es, $P(n_k) = P(n_k | c_k)$.
- ❑ Modelo de bigramas de atributos. Predice cuáles atributos incluir en la intervención. Para disponer de un modelo sencillo, se establecen varios supuestos: a) las distribuciones de probabilidad de los atributos dependen sólo de los atributos en las intervenciones anteriores (y, al limitarse a bigramas, se considera sólo la inmediata anterior); b) los atributos en la intervención previa del usuario, $B = \{b_1, \dots, b_m\}$, contribuyen independientemente a las probabilidades de los atributos en la siguiente intervención del sistema, $A = \{a_1, \dots, a_n\}$; c) los atributos en la intervención del sistema son independientes entre sí. Con estas simplificaciones, el modelo es:

$$A^* = \arg \max \sum_{k=1..m} P(b_k) \prod_{i=1..n} P(a_i | b_k)$$

En la construcción de las frases (*surface realization*) se llevan a cabo cuatro procesos:

- ❑ Construcción de los modelos. Se trata de un proceso previo. Usando las intervenciones etiquetadas de los corpus, se construyeron modelos de n-gramas no suaviza-

dos para cada clase de intervención. Se eligió $n = 5$ como compromiso para dotar de variabilidad a las salidas generadas y evitar la generación de frases sin sentido.

- Generación de las intervenciones (*generating utterances*). En el frame que transmite el gestor de diálogo, los dos primeros pares atributo/valor especifican la clase de intervención. En consecuencia, se activa el correspondiente modelo de lenguaje, y se generan secuencias de palabras según las distribuciones de probabilidad del modelo. El uso de 5-gramas no suavizados garantiza que no se construyan frases sin sentido, al menos para ese tamaño de ventana.
- Puntuación de las intervenciones (*scoring utterances*). Para cada intervención así generada se calcula un valor de penalización, aplicando varios criterios heurísticos como: a) exceso o defecto de longitud, para su clase; b) repetición de contenedores de atributos (*slots*); c) presencia de *slots* para los que hay valor en el frame; d) ausencia de *slots* obligatorios. El procedimiento es iterativo, manteniéndose sólo la intervención mejor puntuada, y terminando cuando se encuentra una sin penalización o vencido un plazo.
- Asignación de valores (*filling slots*).

En (Oh y Rudnicky, 2000) se concluye con los resultados de una evaluación preliminar. La evaluación de los generadores de lenguaje es una cuestión difícil, particularmente en los SDS, dada la dificultad de aislar el comportamiento del módulo generador de respuestas respecto al resto del sistema. La solución más simple, y seguida en (Oh y Rudnicky, 2000), es una evaluación comparativa, donde se prueben dos sistemas idénticos, variando sólo el generador de respuestas.

En lo relativo a planificación de contenidos, se compara la propuesta estocástica con otro algoritmo que, usando la historia del diálogo, selecciona atributos según una sencilla heurística: los atributos se marcan como viejos, si ya han sido enunciados por el sistema, o como nuevos, si aún no se han citado y sólo éstos serán los candidatos a aparecer en las siguientes intervenciones (habitualmente, en la inmediata, salvo si ésta no contiene el oportuno *slot* en su plantilla). En cuanto a la calidad de las frases construidas, se compara con otro generador basado en plantillas.

Los resultados, aunque no concluyentes, indican que los usuarios valoran de modo muy parecido los generadores comparados. Se destaca que sea así, cuando los modelos estocásticos considerados son muy simples, lo que apunta a la posibilidad de claras mejoras.

La aproximación estocástica presenta importantes ventajas: a) en tiempo de respuesta, generando intervenciones a más velocidad que cualquier sistema basado en reglas; b) en coste de implementación, dado que el esfuerzo de etiquetado es sensiblemente menor que la adquisición de conocimiento para los sistemas basados en reglas e, incluso, en plantillas; c) en capacidad de adaptación, a nuevos dominios, o incluso a nuevos idiomas.

En (Oh y Rudnicky, 2002), con más amplitud, se describe la misma metodología y resultados. El sistema de generación de respuestas implementado es, en realidad, un sistema híbrido, resultante de ampliar un anterior sistema basado en plantillas con los métodos estocásticos que se han explicado.

2.5 Evaluación de los sistemas de diálogo

Otra cuestión muy discutida por los investigadores es la del establecimiento de medidas de evaluación de los sistemas de diálogo. En (Dybkjaer et al., 2004) se expone el estado del arte al respecto a fecha de 2004. A diferencia de los componentes de reconocimiento del habla y de comprensión del lenguaje (donde se dispone de métricas ampliamente aceptadas), existen serias dificultades para medir objetivamente el funcionamiento de los gestores de diálogo, y también de los generadores de respuestas. La funcionalidad de estos módulos está estrechamente vinculada a diversos aspectos de su usabilidad y, por ello, no existe un método de evaluación que sea exclusivamente técnico.

Además del modelo PARADISE, en (Dybkjaer et al., 2004) se describen otros proyectos que han aportado métodos genéricos para la evaluación, a partir de sus respectivos resultados experimentales:

- ❑ ATIS. El método consiste en medir la información dada al usuario, comparando la respuesta generada por el sistema con pares de respuestas, mínima y máxima, de referencia previamente anotadas. Entre sus puntos débiles destaca que exige condiciones poco realistas, como mantener una base de datos estática, y que no valora importantes aspectos de la generación del lenguaje, pues considera la respuesta sólo como una tabla de valores.
- ❑ Evalda. Se trata de un proyecto francés, orientado a la evaluación de las tecnologías del lenguaje desarrolladas para ese idioma. En el caso de los sistemas de diálogo, se usan conjuntos de test procedentes de corpus reales, una representación semántica del diálogo y métricas de evaluación comunes. No se han publicado todavía sus resultados.
- ❑ EAGLES (*Expert Advisory Group on Language Engineering Standards*). Se proponen medidas cuantitativas (tasa de finalización de la tarea, tasa de éxito de la transacción, tiempo de respuesta del sistema, concisión de las respuestas del sistema...) y cualitativas (satisfacción del usuario, capacidad de adaptación a nuevos usuarios, capacidad para manejar la multimodalidad...). Además de proponer qué evaluar, se busca también establecer cómo evaluar e informar de los resultados, fijando un conjunto de parámetros (del sistema, de condiciones del test, de resultados del test) que permitan una comparación homogénea.
- ❑ DISC (*Spoken Language Dialogue Systems and Components*). En la misma línea que el anterior proyecto, se propone qué evaluar (el conjunto de propiedades) y cómo evaluar (el criterio a aplicar). La metodología se basa en el uso de plantillas y ciclos de vida.

De la experiencia de DISC y del proyecto danés de diálogo, los autores de (Dybkjaer et al., 2004) extraen una lista de aspectos relevantes para juzgar la usabilidad de los SDS: a) exactitud del reconocimiento de la entrada; b) naturalidad del habla del usuario; c) calidad de la voz de salida; d) adecuación de las frases de salida; e) adecuación en informar al usuario de los datos comprendidos y de las acciones en curso (*feedback adequacy*); f) adecuación en la iniciativa del diálogo; g) naturalidad en la estructura del diálogo; h) suficiente cobertura del dominio de la tarea; i) suficiente capacidad de razonamiento (interpretación del contexto);

j) suficiencia en la guía de interacción (mecanismos de ayuda); k) adecuación en el manejo de errores; l) suficiencia en la adaptación a las diferencias de los usuarios; m) propiedad de la modalidad (en el caso de los sistemas multimodales).

- PARADISE (Walker et al., 1997)

Entre las propuestas realizadas destaca la denominada PARADISE (Walker et al., 1997). El modelo PARADISE (*Paradigm for Dialogue System Evaluation*) combina diferentes medidas en una única función que mide el rendimiento del sistema, en correlación directa con la satisfacción de los usuarios del sistema. A este método de evaluación están recurriendo diferentes investigadores para medir la calidad de sus sistemas, como es el caso del PARIS-SITI, tal como se expone en (Bonneau-Maynard et al., 2000) y (Devilleers et al., 1998).

Según consta en (Dybkaer et al., 2004), los supuestos principales del modelo PARADISE son: a) el objetivo a maximizar es la satisfacción del usuario; b) el éxito en la tarea y varios costes asociados a la interacción pueden usarse para predecir la satisfacción del usuario.

En (Dybkaer et al., 2004) también se informa de las debilidades potenciales del modelo: a) excesivo acoplamiento entre satisfacción de usuario y usabilidad; b) dudas acerca de la posibilidad de predecir la satisfacción de usuario a partir de la información registrada en los ficheros *log* del sistema; c) dificultad en la interpretación de los cuestionarios, y ausencia de fundamentos teóricos acerca de las cuestiones a incluir en los mismos; d) uso del modelo, por el momento, limitado a experimentos controlados (y no con usuarios reales).

En (Walker et al., 1997) se presenta el modelo PARADISE como un marco general para la evaluación de los agentes de diálogo, dado que permite desacoplar los requisitos de la tarea y los comportamientos de los agentes, comparar estrategias de diálogo, calcular el rendimiento de diálogos completos y de subdiálogos, especificar las contribuciones relativas al rendimiento, y comparar agentes que aborden diferentes tareas, mediante una normalización de la complejidad de la tarea.

Según se explica en (Walker et al., 1997), la metodología de PARADISE consta de los siguientes pasos:

- ❑ Definición de una tarea y un conjunto de escenarios.
- ❑ Especificación de la tarea mediante AVM (*Attribute Value Matrices*), en las que se especifica la información que debe intercambiarse entre agente y usuario durante el diálogo.
- ❑ Experimentación con los distintos agentes de diálogo para la tarea.
- ❑ Cálculo de la satisfacción de los usuarios usando cuestionarios.
- ❑ Cálculo del éxito en la tarea usando el coeficiente Kappa (calculado a partir de matrices de confusión, en las que se comparan las AVM del escenario y del diálogo realizado).
- ❑ Cálculo del coste del diálogo usando medidas de eficiencia y cualitativas.

- ❑ Estimación de una función de rendimiento usando regresión lineal y los valores de satisfacción del usuario, Kappa y los costes del diálogo.
- ❑ Comparación con otros agentes y/o tareas para determinar los factores a generalizar.
- ❑ Refinamiento del modelo de rendimiento.

En (Walker et al., 1998), además de describir el citado modelo PARADISE, se ilustra su aplicación para evaluar dos sistemas de diálogo hablado. Entre las ventajas de este modelo de evaluación, se tiene que:

- ❑ Combina, en una única función de rendimiento, medidas de éxito en la tarea y medidas, objetivas y subjetivas, de los costes del diálogo. El objetivo principal es maximizar la satisfacción del usuario y, para ello, se buscará maximizar el éxito en la tarea y minimizar los costes del diálogo.
- ❑ Permite comparar varias estrategias de diálogo, y, especialmente, dado que las medidas de rendimiento se normalizan respecto a la complejidad de la tarea, permite comparar varios agentes que realicen diferentes tareas.
- ❑ Permite evaluar el rendimiento en cualquier nivel del diálogo, puesto que el éxito de la tarea y los costes de diálogo pueden calcularse también para cualquier sub-tarea del diálogo.

Otro aspecto, algo marginal en (Walker et al., 1998) pero interesante, es el uso de la función de rendimiento de PARADISE como una realimentación del agente para que éste pueda, automáticamente, aprender las estrategias óptimas en aquellas situaciones de diálogo en las que disponga de varias estrategias.

- LIMSI (Lamel et al., 2000)

En (Lamel et al., 2000a) se informa de la evaluación del sistema ARISE exclusivamente en términos de éxito en el diálogo (*dialog error rate*) y duración, en segundos, de las llamadas.

En cambio, en (Devillers et al., 1998), los investigadores del LIMSI sí usan la metodología PARADISE al informar de una evaluación comparativa de dos diferentes estrategias de diálogo, probadas sobre el mismo sistema de información turística, PARIS-SITI. Las estrategias comparadas difieren en que una dispone de un mecanismo automático para guiar al usuario mediante sugerencias (*prompts* no directivos), mientras que la otra carece de tal funcionalidad. Se considera que PARIS-SITI es un sistema con un rango de posibilidades de consulta más amplio que el de ARISE. Por ello, el objetivo de la estrategia con sugerencias es ayudar al usuario a descubrir las posibilidades del sistema, sugiriéndole términos válidos del léxico de la tarea, pero sin llegar a las restricciones de las estrategias de iniciativa del sistema.

Los sujetos partícipes en la evaluación fueron 16 usuarios expertos y 16 usuarios novatos. Se esperaba que la estrategia con sugerencias fuera preferida por todos, pero que resultara más ventajosa para los novatos. La evaluación confirmó tal hipótesis, pues los novatos tendieron a seguir las sugerencias del sistema más que los expertos (66.3% frente a 51% de

las veces). Además, los usuarios que probaron antes el sistema con la estrategia con sugerencias no advirtieron la diferencia al probar la otra variante, mientras que los que probaron primero la estrategia sin sugerencias sí percibieron el cambio y, claramente, prefirieron la estrategia con sugerencias. En términos de PARADISE, la medida de éxito en la tarea, Kappa, mostró también la superioridad de esta estrategia (0.967 frente a 0.869).

En (Bonneau-Maynard et al., 2000), se trata de nuevo la evaluación del PARIS-SITI y de ARISE, desde el paradigma de PARADISE. El objetivo de esta experimentación es estudiar las posibilidades de predecir la satisfacción de los usuarios a partir de las medidas de coste del diálogo y éxito de la tarea. Se observó que los pesos asociados a los distintos factores que contribuyen a la función del rendimiento del sistema se mantienen bastante estables, cuando se cambia de conjunto de sujetos evaluadores, dentro de la misma tarea, e incluso cuando se cambia de dominio (de PARIS-SITI a ARISE). En consecuencia, la función del rendimiento del sistema podría usarse para realizar predicciones acerca de la evolución futura de los sistemas sin necesidad de llevar a cabo amplias y costosas interacciones con los usuarios.

- *(Baggia et al., 2000), (Sturm et al., 1999)*

En (Baggia et al., 2000), se presentan los resultados de dos evaluaciones del prototipo italiano de este proyecto (CSELT ARISE), integrado en el centro de atención al cliente de los ferrocarriles italianos (FS-Infoma) y atendiendo llamadas de todo tipo de usuarios. De estas pruebas de campo, se recopiló la valoración de los usuarios que cumplimentaron los oportunos cuestionarios (que constaban de 12 ítems puntuables en una escala de 5 valores). Esta evaluación subjetiva se complementa con otra, objetiva, donde, como en (Lamel et al., 2000a), se informa del porcentaje de diálogos con éxito y de la duración media de estos diálogos. Finalmente, dada la integración del prototipo en un entorno real de trabajo, se destaca que el número medio de llamadas atendidas al mes pasó de 660.000 a 1.500.000, tras integrar el SDS (*spoken dialogue system*) en el centro y mantener el mismo número de operadores humanos. Además, el sistema está operativo las 24 horas del día.

En (Sturm et al., 1999a), se informa de la evaluación del prototipo holandés de ARISE. Al igual que en otros trabajos, se combina una evaluación objetiva (rendimiento del reconecedor, tasa de éxito del diálogo, número de turnos) con otra subjetiva (respuestas de los usuarios a un cuestionario de 7 preguntas abiertas). Realizaron las pruebas 25 expertos en SDS y 200 usuarios anónimos, menos familiarizados con estos sistemas.

Recientemente, en (Sturm et al., 2002), el mismo grupo de investigadores presenta una evaluación de un prototipo de sistema multimodal (integración de habla e interfaz gráfica), desarrollado en el proyecto MATIS (*Multimodal Access to Transaction and Information Services*). Como en anteriores referencias, la comparación entre el nuevo sistema multimodal y el SDS se formula en términos de tasa de éxito en los diálogos y duración en segundos.

- AT&T (Eckert et al., 1997)

En (Eckert et al., 1997), los investigadores de AT&T proponen el uso de un usuario simulado como metodología de evaluación automática que permita, además, optimizar la estrategia en el nivel de diálogo. Con este tipo de evaluación, se dispondría de un número elevado de diálogos, sin llevar a cabo una costosa investigación manual, y los resultados serían más consistentes y menos proclives a errores. Por otra parte, un mismo modelo de usuario podría usarse para una evaluación comparativa de varios sistemas de diálogo que aborden una tarea dada.

- CMU (Rudnicky et al., 1999)

En (Rudnicky et al., 1999b), como componente del sistema modular CMU Communicator, se cita a un supervisor humano. En ausencia de problemas, el sistema supervisor se limita a una monitorización pasiva. Cuando los problemas se presentan (y se detectan automáticamente mediante razonamiento basado en reglas), se transferirá el control al supervisor humano. Este procedimiento permite comprender mejor las situaciones de error y caracterizar estrategias razonables de recuperación de errores, estrategias que, más tarde, podrán ser automatizadas e incorporadas al sistema.

En (Constatinides y Rudnicky, 1999), se describe con más detalle el procedimiento de detección de fallos en el CMU Communicator. Se pretende sistematizar el proceso de análisis de los diálogos. Para alcanzar este objetivo, se plantean las siguientes tareas: a) descubrir todas las instancias de fallos de interacción; b) desarrollar un procedimiento de anotación de fallos, que proporcione una documentación eficiente; c) especificar el procedimiento de manera que pueda ser aplicado por un asesor que no conozca los detalles de diseño del sistema.

La detección de los fallos se facilita con el uso de una herramienta que compara los objetivos de usuario y sistema en cada turno de diálogo. En la fase de diagnóstico, se dispone de una categorización visual de las fuentes de fallo del sistema. Se usan herramientas de generación automática de informes para documentar los errores. El resultado, para cada sesión de diálogo, es una tabla donde cada fila informa de un error: número de turno, categoría del error, y una breve descripción del error. Además, por cada sesión se añade otra fila con el resultado de la misma: a) sesión completada con itinerario inconsistente con las necesidades del usuario; b) sesión terminada prematuramente por el usuario; c) sesión terminada prematuramente por el sistema; d) sesión completada satisfactoriamente.

La metodología presentada permite a los asesores realizar análisis cualitativos, descubriendo fallos que el sistema no puede detectar automáticamente, y caracterizar y documentar estos fallos, de modo preciso. Esta documentación servirá a los diseñadores del sistema para comprender mejor los aspectos que necesitan mejora.

2.6 Simulación de diálogos

Un serio problema en el perfeccionamiento de los sistemas de diálogo es el elevado coste que representa su evaluación real con usuarios humanos. En consecuencia, se vienen discutiendo diferentes propuestas, (Eckert et al., 1997), (López-Cózar et al., 2003) y (Scheffler y Young, 2001), acerca del uso de diálogos simulados que permitan el aprendizaje o depuración de los modelos y la evaluación de los sistemas.

Ante el desarrollo de esta línea de investigación por parte de distintos grupos, se ha planteado también la necesidad de establecer criterios de evaluación de los modelos o técnicas de simulación propuestos. Una panorámica de algunas técnicas de simulación y una primera propuesta para su evaluación cuantitativa se ha presentado en (Schatzmann et al., 2005).

- *AT&T (Eckert et al., 1997)*

En (Eckert et al., 1997) y (Levin et al., 2000a) se justifica la necesidad de programar un usuario simulado por el elevado número de iteraciones requeridas en el aprendizaje automático de la estrategia óptima de diálogo. La simulación del usuario resulta especialmente adecuada en las primeras etapas de dicho aprendizaje, cuando la estrategia es tan deficiente que los usuarios reales sentirían un rechazo completo a proseguir la sesión. Más tarde, una vez la interacción con el usuario simulado proporcione una estrategia razonable de diálogo, el aprendizaje podrá continuar mediante sesiones con usuarios reales.

Para este primer estadio del aprendizaje, el usuario propuesto es un modelo estocástico generativo (obtenido por aprendizaje supervisado a partir del corpus) que produce actos de diálogo en respuesta a las acciones de diálogo del sistema. Se considera que la libertad de interacción o iniciativa de los usuarios puede modelarse mediante distribuciones de probabilidad. Estas distribuciones dependen sólo de las respuestas previas del sistema y de la descripción del estado de usuario. En (Eckert et al., 1997), se restringe el modelo propuesto a decidir según una historia de longitud uno, es decir, la respuesta del usuario dependerá sólo del turno precedente del sistema:

$$p_{ij} = p(U_t = I_i | S_{t-1} = I_j)$$

Se podrían incorporar contextos de más longitud al modelo, sin embargo la escasez de datos de entrenamiento lo desaconseja. Existen, además, otros parámetros en el modelado del usuario relacionados con convenciones de las conversaciones como el número de turnos que llevan al usuario a colgar el teléfono (límite de la paciencia del usuario) o el número de datos facilitados espontáneamente en la primera intervención del usuario.

En (Eckert et al., 1997), se citan los tres niveles de interacción entre usuario y sistema (señales acústicas, secuencias de palabras, e intenciones) y se restringe el modelo de usuario al nivel intencional, descrito mediante actos de diálogo. En consecuencia, con el usuario así simulado no hay riesgo de malentendidos.

Se plantea, entre las futuras mejoras, la posibilidad de modelar al usuario a nivel de palabras. De esta manera, se podría simular el efecto de diferentes tasas de reconocimiento de

palabras o errores típicos de reconocedores concretos, abriendo nuevas posibilidades de evaluación de los sistemas.

Este usuario simulado también es útil para probar el sistema, en amplitud y con coste mínimo, y detectar fallos o errores de estrategia.

- Universidad de Granada (López-Cózar et al., 2001)

Investigadores de la Universidad de Granada plantean, en (López-Cózar et al., 2001), (López-Cózar et al., 2002) y (López-Cózar et al., 2003), la generación automática de diálogos como una herramienta eficaz en la depuración y prueba de los sistemas de diálogo, antes de habilitar el acceso de usuarios reales al mismo. Han aplicado esta técnica en el sistema SAPLEN, cuyo objetivo es atender telefónicamente los pedidos de los clientes de restaurantes de comida rápida.

En (López-Cózar et al., 2001), se presenta un módulo simulador de usuario que facilita la evaluación y ajuste del sistema ante diversas condiciones de ruido. El simulador debe ser capaz de: a) conocer todos los posibles tipos de *prompts* del sistema; b) ejecutar acciones que generen respuestas apropiadas a los mismos. Existen acciones a definir que son dependientes del dominio de la aplicación, pero también hay dos tipos de acciones comunes a todos los dominios: la recuperación de errores y la confirmación.

Además, para poder usar el simulador, se requiere construir dos corpus: a) un corpus de escenarios de diálogo, que represente los posibles objetivos de los usuarios, que se codificarán mediante frames; b) un corpus de frases, grabadas por distintos locutores, que sean adecuadas para abordar los escenarios considerados. De este modo, la estrategia del simulador, orientada a alcanzar objetivos, depende del corpus de escenarios. Fijada una respuesta del simulador, en forma de representación semántica, se accede al corpus de frases y se elige y emite una del subconjunto de frases que corresponda a ese significado (la existencia de varias alternativas para cada significado proporciona flexibilidad a la generación de los diálogos).

Se informa de una experimentación para evaluar el sistema ante diferentes niveles y tipos de ruido (ruido blanco y ruido *babble* o de conversaciones de fondo) en la que se han recopilado 28.800 diálogos (100 diálogos por cada uno de los 18 escenarios y cada una de las 16 condiciones de nivel, tipo y tratamiento del ruido). La métrica de evaluación es el logro de los objetivos de los escenarios, pero dentro de un margen de duración máxima de los diálogos (se fijó que el simulador cancelara la sesión tras 30 interacciones aunque le quedaran objetivos por cumplir).

Aparte de conclusiones relativas al tratamiento del ruido, la simulación permite detectar errores en la estrategia de confirmación implementada: a) confirmaciones negativas que el ASR transforma en positivas, con la consiguiente aceptación de datos incorrectos; b) confirmaciones positivas, mudadas en negativas por el ASR, obligando a turnos adicionales para confirmar el dato. Se apunta como mejora a estudiar en un futuro el uso de confirmaciones redundantes, aunque con las reservas que implica en incremento de la duración de los diálogos.

En (López-Cózar et al., 2002) y (López-Cózar et al., 2003), se describe con más detalle el sistema SAPLEN y el simulador de usuario. La experimentación reportada en (López-Cózar et al., 2002) es la misma que en (López-Cózar et al., 2001), pero también descrita en amplitud.

En (López-Cózar et al., 2003), la misma técnica se aplica para, entre otros fines, experimentar diferentes estrategias de confirmación. En una nueva experimentación que consta de 1.800 diálogos (100 diálogos por cada uno de los 18 escenarios), se apreció que 130 no terminaban satisfactoriamente (sea porque el sistema no entendió correctamente parte del objetivo del escenario, sea porque la sesión superó el límite de interacción, 30 turnos, y el simulador del usuario la canceló). Estos diálogos fallidos se deben a datos incorrectamente confirmados (a causa de errores del ASR que cambian confirmaciones positivas por negativas, y viceversa).

Se propone una nueva estrategia de confirmación con repetición del dato (*re-prompting confirmation*). En lugar de pedir una respuesta “sí/no”, característica de la confirmación explícita, se solicita de nuevo el mismo elemento de información. El procedimiento se repite hasta que el sistema identifique el mismo valor en dos turnos seguidos.

Al repetir los mismos experimentos siguiendo esta nueva estrategia se vio su clara superioridad (98.22% de cumplimiento de escenario con *re-prompting confirmation* frente a 92.77% usando confirmación explícita). Este tipo de confirmación con repetición supone un modo más robusto de asegurar la validez de los datos, pues se basa en la comparación de representaciones semánticas, mientras que la confirmación explícita depende de la correcta identificación de una palabra aislada.

Finalmente, en (López-Cózar et al., 2003), se plantean limitaciones de esta técnica de simulación e ideas para continuarla. El uso de un simulador, programado de modo que siempre facilita la información solicitada (sólo la adición de errores del ASR causa que no se alcance el 100% de éxito), y que nunca plantea dudas, ni se siente confundido o nervioso, puede juzgarse como un entorno de experimentación muy poco realista. Aún así, es útil para probar y ajustar, a muy bajo coste, distintos aspectos de los sistemas, como las citadas variantes en la estrategia de confirmación. Como cuestiones abiertas para futura investigación se plantea: a) inclusión de nuevos tipos de objetivo en los escenarios, considerando, por ejemplo, que el usuario pueda cambiarlo, dinámicamente; b) refinamiento de la técnica para incluir más datos en los escenarios y obtener mejores modelos de los usuarios reales; c) uso de probabilidades, asociadas a los objetivos, que permitan variar el modo en que el simulador responde a determinados *prompts*, considerando incluso la posibilidad de que aporte información errónea; d) establecimiento de límites de interacción no sólo para el diálogo completo, sino también para los subdiálogos, como medio para prevenir situaciones anómalas donde el sistema pida repetidamente el mismo dato y el simulador, pacientemente, se lo proporcione, sin cancelar la sesión (como haría cualquier usuario humano).

- *University of Cambridge (Scheffler y Young, 2001)*

En (Scheffler y Young, 2001), se describe un método para simular diálogos de iniciativa mixta a partir del corpus de diálogos recopilado por un prototipo del sistema. Las intervenciones de los usuarios en el corpus permiten modelar estocásticamente el simulador de los

usuarios. Este sistema se aplica para comparar resultados de diferentes estrategias de diálogo así como para el aprendizaje automático de las estrategias.

El método propuesto para llevar a cabo la simulación de diálogos consiste en modelar, a partir de un corpus, el comportamiento del usuario y los errores del sistema. Respecto al usuario, el principal supuesto es que siempre actúa con intención de completar ciertos objetivos (*goal directed user model*) y que cada objetivo se mantiene como tal hasta que es satisfecho. Los conceptos y estructuras principales de este modelo son:

- ❑ El objetivo (*goal*), que se representa mediante una estructura de datos atributo-valor y, además, a cada atributo del objetivo se le añade una variable de estado, que puede tomar los valores: a) pendiente, si aún no se ha facilitado; b) especificado, cuando ya se ha dado y no debe repetirse; c) urgente, si es prioritario especificarlo, por ejemplo si se cree que el sistema no lo ha entendido bien; d) no aplicable, si no debe especificarse en la transacción considerada.
- ❑ Las intenciones (*intention language*), que se definen como las unidades mínimas de información que pueden aportarse de modo independiente. Existen intenciones asociadas a etiquetas semánticas específicas, y otras especiales como las de tipo *content*, que sólo indican si el valor es correcto o incorrecto, de tipo *null*, para indicar la ausencia de cierta intención, y de tipo *mumble*, para representar información ininteligible, como la que esté fuera del vocabulario del sistema.
- ❑ Las redes de generación de intervenciones (*utterance generation lattices*), cuyos arcos se corresponden con las intenciones y cuyos nodos son puntos de elección, entre las intenciones disponibles. Para generar las frases del usuario, se construye un camino a través de la red: a) se identifican los arcos permitidos según las variables de estado de los atributos que componen el objetivo; b) se resuelven los nodos, en unos casos de modo probabilístico y en otros determinista; c) en la opción elegida, se puede introducir error, realizando una sustitución probabilística de la intención seleccionada que puede ocasionar un salto a otro arco de la red.

En (Scheffler y Young, 2001) se describe cómo se ha realizado el entrenamiento de este tipo de modelado del usuario a partir del corpus. Las probabilidades del modelo se estiman mediante contadores de ocurrencia de sucesos en una situación o contexto determinado y contadores de ocurrencia de los contextos: $P(\text{Event} \mid \text{Context}) = \text{cnt}(\text{Event}, \text{Context}) / \text{cnt}(\text{Context})$.

En una experimentación ideal, se desearía contrastar los resultados de la simulación con diálogos mantenidos con usuarios reales, usando el mismo sistema. Se plantearía comparar en términos de cumplimiento de los objetivos del usuario, pero éstos en el caso de los usuarios reales no se conocen (salvo si se plantea una experimentación con escenarios prefijados, con el consiguiente problema de la falta de naturalidad en los resultados). Por todo ello, se prefiere informar de resultados experimentales comparando los tiempos necesarios para completar los objetivos (es decir, duración del diálogo, número de turnos) en las simulaciones y en los datos de test (parte del corpus no usada en el entrenamiento).

En el análisis de los resultados, se distingue entre diferentes tipos de objetivos. En promedio, los objetivos más simples (que requieren diálogos más breves) se satisfacen con igual número de turnos en el test y en la simulación. En cambio, la simulación resulta ser más deficiente ante los objetivos más complejos (en los que se debe informar al sistema de va-

rias combinaciones de campos objetivo y, por tanto, se requieren diálogos más extensos). En estos casos, la simulación necesita, en promedio, de uno o dos turnos más que los diálogos del test. Finalmente, se observa que establecer el contexto es un factor muy importante para mejorar en este tipo de diálogos, pues se reduce la duración, aunque sigue superando apreciablemente la media del conjunto de test.

En (Schatzmann et al., 2005), se exponen los principales modelos de simulación de usuario basados en el nivel intencional del diálogo y en métodos probabilísticos (es decir, en el modelado del usuario a partir de un corpus de diálogos):

- ❑ Modelo de bigramas. Es el modelo original de (Eckert et al., 1997). El modelo predice la acción del usuario, a_u , como respuesta a una acción del sistema, a_s , es decir: $p = P(a_u|a_s)$.
- ❑ Modelo de Levin. Es una versión mejorada del anterior modelo, que incluye restricciones de las acciones del usuario con el fin de mejorar la cooperación del usuario simulado así como la consistencia de los diálogos. Así, ante una petición de un atributo A_x , por parte del sistema, se modela la probabilidad de que el usuario simulado proporcione A_x , $P(\text{provide } A_x \text{ } | \text{request } A_x)$, y también otros n atributos, $P(n| \text{request } A_x)$.
- ❑ Modelo de Scheffler. Es el modelo de (Scheffler and Young, 2001). Es un modelo mixto, donde se combina lo probabilístico, en el modelado del comportamiento conversacional, y lo determinista, al usar reglas que fijen las acciones dependientes de los objetivos del diálogo. Estas reglas de consistencia resuelven la falta de planificación de los objetivos que se da en los anteriores modelos.
- ❑ Modelo de Pietquin. El modelo de (Pietquin, 2004) amplía el modelo probabilístico de Levin con la representación explícita del objetivo del usuario: $P(\text{provide } A_x | \text{request } A_x, \text{ goal})$. Como en el modelo de Scheffler, los objetivos del usuario se representan mediante una tabla de pares atributo-valor. Una ventaja, respecto a Scheffler, es que el modelo de Pietquin es independiente del dominio. Una importante desventaja es que no se ha obtenido a partir de un corpus, sino que las probabilidades se han fijado manualmente, atendiendo a criterios de sentido común.

A continuación, en el mismo artículo, se describe una metodología para la evaluación cuantitativa y comparativa de las distintas técnicas de simulación. Se consideran dos aspectos o niveles de detalle en la evaluación:

- ❑ Comparación de respuestas aisladas. Consiste en comparar aisladamente las respuestas dadas por usuarios reales y usuarios simulados. Esto permite determinar si el modelo de usuario puede generar intervenciones parecidas a las reales. En la evaluación del artículo, se miden *precision* y *recall* respecto a las acciones de diálogo presentes en un conjunto de test de diálogos reales.
- ❑ Comparación de diálogos. En este caso, se comparan propiedades relevantes de diálogos completos adquiridos por ambos métodos (usuarios reales y simulados). En este caso, el objetivo es estudiar si la simulación puede reproducir la variedad de comportamiento de los usuarios reales (en lugar de un usuario promedio). Se informa de las siguientes propiedades: longitud del diálogo (número de turnos), longitud del turno (número de acciones de diálogo por turno), índice de participación

(ratio de acciones de sistema y usuario por diálogo), frecuencia de las diferentes acciones de sistema y usuario, índices de cooperación (según frecuencias de que las unidades de información sean solicitadas, suministradas, una vez o más veces...), tasas de éxito (ratio de cumplimiento de objetivos o sub-objetivos de diálogo), etc.

En la experimentación de (Schatzmann et al., 2005), se comparan diálogos simulados mediante tres modelos (bigramas, Levin y Pietquin) con cuatro corpus de diálogos reales (adquiridos usando sistemas de ATT, BBN, CMU y SRI). Se concluye que el modelo básico de simulador de usuario (el modelo de bigramas) no genera diálogos de suficiente calidad. En cambio, los modelos de Levin y Pietquin lo superan claramente, proporcionan una simulación de calidad, aunque los diálogos simulados pueden distinguirse fácilmente de los diálogos reales mediante sencillas medidas estadísticas (como las enumeradas anteriormente).

2.6.1 Adaptación a usuario y/o dominio

Existe también interés en el desarrollo de sistemas de diálogo capaces de proporcionar información de diferentes dominios, o capaces de adaptarse a las preferencias de sus usuarios.

- *(Pargellis et al., 2004)*

En (Pargellis et al., 2004) se describe una arquitectura modular para un sistema de diálogo en el que se integra un generador de aplicaciones (AG: *Application Generator*). Este módulo AG consta de los siguientes componentes:

- ❑ un gestor de perfiles (PM: *Profile Manager*), que almacena las preferencias de los usuarios;
- ❑ un gestor de información (ISM: *Information and Services Manager*), que accede a las bases de datos externas para extraer y procesar la información conforme a las preferencias de los usuarios;
- ❑ un generador automático de diálogos (ADG: *Automatic Dialogue Generator*), que, dada una especificación tabular de cierta tarea, genera una estructura de diálogo para esa tarea, mediante un grafo que conecta un conjunto finito de estados de diálogo (nodos del grafo);
- ❑ un gestor de diálogo (DM: *Dialogue Manager*), encargado del mantenimiento de la sesión de diálogo con el usuario.

Este módulo AG permite crear aplicaciones adaptadas a las preferencias del usuario, modificando la estructura del diálogo (por ejemplo, suprimiendo turnos de pregunta o confirmación de datos, al considerar los datos disponibles en el fichero de preferencias seleccionado por el PM). Además, facilita los cambios en la gestión de la tarea considerada o, incluso, la migración a otras tareas, dado que la especificación de las mismas se encuentra en una serie

de librerías y el ADG es un módulo genérico, que lee estas librerías, entre otras fuentes de información.

- MIT (Glass y Weinstein, 2001)

En (Glass y Weinstein, 2001), los investigadores del MIT presentan una utilidad, llamada Speech-Builder, diseñada para que los programadores de sistemas de diálogo hablado puedan especificar fácilmente la información lingüística propia de los dominios modelados, de manera que se acelere la creación de estos sistemas para nuevas tareas.

En reconocimiento del habla, Speech-Builder permite editar las pronunciaciones. En comprensión, configurar un fichero de gramática y un fichero de conversión de análisis, totales o parciales, a representaciones semánticas. El gestor de diálogo parece ser un componente más difícil de tratar con esta utilidad, dado que suele ser adaptado más ampliamente para cada dominio. Por ello, con Speech-Builder se crea un gestor genérico, enfocado a manejar los tipos de situaciones que se plantean en las consultas a bases de datos.

Con la herramienta Speech-Builder se han desarrollado algunos prototipos, que no han sido sometidos a una evaluación rigurosa. Se tiene intención de aplicar la metodología para configurar un sistema de información meteorológica y evaluar los componentes de reconocimiento y comprensión. Se pretende demostrar que el rendimiento es similar al de los sistemas diseñados manualmente, sin la ayuda de la herramienta.

- AT&T (Eckert et al., 1997)

También, tal como se plantea en (Eckert et al., 1997), el uso de usuarios simulados, parametrizados convenientemente, constituye un método eficiente para evaluar los SDS y adaptar las estrategias. Aquí se plantearon cuatro tipos de usuario para la tarea ATIS:

- Usuario de referencia: probabilidades del modelo fijadas según sentido común.
- Usuario paciente: que sólo colgaría después de 99 turnos, y cuyos diálogos siempre deberían terminar satisfactoriamente.
- Usuario sumiso: que responde siempre a las preguntas del sistema, pero nunca aporta información espontánea.
- Usuario experimentado: mejor conocimiento de la tarea que el usuario de referencia, capaz de aportar información espontánea, y con una paciencia ligeramente alta.

Estos usuarios se conectaron al sistema AMICA y, para cada uno, se simularon 10.000 diálogos. Se apreció que, debido al proceso estocástico, pequeñas variaciones de los parámetros de los perfiles de usuario causan sólo pequeñas variaciones en los resultados de la evaluación. Por tanto, la determinación exacta de tales parámetros no es una cuestión crítica.

Del análisis de los resultados, se obtienen modificaciones para mejorar la estrategia de diálogo. Así, por ejemplo, se observó que, en una gran cantidad de diálogos largos con los modelos de usuario paciente y sumiso, se repetía la misma situación: el sistema informaba

de que no habían datos (vuelo o conexión inexistente, no disponible) y el usuario insistía en solicitar dicha conexión. La modificación de estrategia propuesta es que el sistema termine el diálogo en tal situación.

3. Tarea BASURDE

3. Tarea BASURDE

3.1 Introducción

La tarea definida en el proyecto BASURDE (Bonafonte et al., 2000) consiste en atender consultas telefónicas sobre horarios, precios y servicios de trenes españoles de largo recorrido. Los usuarios del sistema que realice esta tarea formularán sus peticiones en español y el sistema tendrá que responderles en ese mismo idioma. La tarea, por tanto, es similar a la elegida por otros grupos de investigación europeos, por ejemplo, en el proyecto ARISE, desarrollado por el LIMSI, entre otros, o Philips en el proyecto TABA. Este tipo de tareas, respecto a otras, presenta varias ventajas:

- ❑ Se trata de una tarea de semántica restringida y que suele tener un vocabulario estable y regular, lo que permite abordar adecuadamente la comprensión del lenguaje natural.
- ❑ Proporciona una estructura de diálogo (secuencia de interacciones entre usuario y sistema) que tiene una cierta complejidad, de manera que en la gestión del diálogo por parte del sistema se plantearán dificultades para representar esta estructura compleja y diseñar estrategias para tomar las decisiones más adecuadas en cada diálogo.

3.2 Adquisición del corpus

Dado que se consideró la aproximación estadística en el desarrollo del sistema de diálogo, se adquirió un corpus de diálogos. A partir de este corpus es posible el aprendizaje de modelos de lenguaje específicos para la tarea. La adquisición se realizó en dos fases:

- ❑ Adquisición de un corpus de diálogos persona – persona. Se trata de 204 diálogos correspondientes a consultas reales efectuadas en el servicio de información de RENFE. Es un corpus bastante variado que se usó para definir la semántica de la tarea y acotar los tipos de consulta que deberían tratarse en el sistema BASURDE (así, por ejemplo, se excluyeron las consultas relativas a trenes de cercanías). Sin embargo, los diálogos de este corpus presentan una fluidez que no se dará cuando los usuarios hablen con un sistema y, en consecuencia, no conviene usarlos en el aprendizaje de modelos estadísticos.
- ❑ Adquisición de un corpus de diálogos usando la técnica del Mago de Oz (Fraser y Gilbert, 1991). Se trata, también, de diálogos persona – persona, pero con la importante diferencia de que la persona que actúa como usuario desconoce la existencia de la persona que actúa como el sistema (a quien se llama mago) y cree que le atiende un sistema totalmente automático. De este modo, si el mago consigue fingir bien su simulación del sistema, en los diálogos adquiridos se reflejará el comportamiento de los usuarios ante un sistema automático. Por tanto, este corpus será

adecuado para aprender modelos estadísticos de los usuarios que serán, más tarde, utilizados en el sistema BASURDE.

Para efectuar la adquisición mediante Mago de Oz en condiciones controladas, es necesario definir un conjunto de escenarios que establezcan los objetivos que los usuarios o informantes han de procurar satisfacer en sus llamadas y delimitar las situaciones que constituyen el contexto de tales objetivos. En el proyecto BASURDE (Mariño et al., 1999), a partir del análisis del corpus de diálogos persona – persona, se definieron cuatro tipos de escenarios:

- ❑ Escenarios de tipo A. Consulta de horarios y, opcionalmente, tipos de trenes. Sólo se consideran viajes de ida. El informante suministra la ciudad de destino y, opcionalmente, puede facilitar la ciudad de origen, la fecha del viaje (o intervalo de fechas) y el tipo de tren. La ciudad de origen puede no proporcionarse si se considera, implícitamente, que es la ciudad desde donde se efectúa la llamada. La fecha del viaje puede no proporcionarse si se consultan horarios entre dos ciudades para cualquier día (es decir, el informante quiere conocer los horarios, sin necesidad o preferencia por viajar un día determinado).
- ❑ Escenarios de tipo B. Consulta de horarios, precios y, opcionalmente, tipos de trenes. Sólo se consideran viajes de ida. El informante suministra la ciudad de destino y, opcionalmente, puede facilitar la hora del viaje, la ciudad de origen, la fecha del viaje (o intervalo de fechas) y el tipo de tren. Respecto a los escenarios de tipo A, se observan dos diferencias: se incluye la consulta de precios y se permite que el informante pueda especificar la hora (o intervalo de horas) en la que desea viajar.
- ❑ Escenarios de tipo C. Los objetivos de consulta y los datos a suministrar son los mismos que en los escenarios de tipo B. La diferencia reside en que en éstos se consideran viajes de ida y vuelta.
- ❑ Escenarios de tipo D. Escenario libre, totalmente abierto, donde el informante escoge tanto los objetivos a consultar como la situación de contexto que determina tales objetivos.

En los escenarios de tipos A, B y C, se definieron variantes cerradas (en las que todas las variables que definen la situación del escenario tienen sus valores determinados) y variantes semiabiertas (en las que se permite al informante dar valores a las variables).

Se adquirieron 227 diálogos usando la técnica del Mago de Oz. En esta adquisición participaron 75 informantes y cada uno completó 3 escenarios (dos de los tipos A, B o C, y uno de tipo D). En la tabla 3.1 se resumen los datos más relevantes del corpus adquirido:

Nº diálogos	227	50 tipo A, 51 tipo B, 51 tipo C, 75 tipo D
Nº turnos	3.528	1.654 turnos de usuario, 1.874 turnos de sistema
Nº palabras	14.902	637 palabras distintas

Tabla 3.1. Estadística del corpus BASURDE

3.3 Etiquetado del corpus: representación semántica mediante frames

Antes de aprender modelos de lenguaje a partir del corpus adquirido, es preciso proceder a su etiquetado. Se tuvo que definir un conjunto de etiquetas que describiera la semántica de la tarea, un conjunto de reglas que estableciera cómo etiquetar las diferentes unidades o elementos que forman los diálogos, y finalmente aplicarlas, etiquetando los 227 diálogos del corpus (Martínez et al., 2002).

A fin de obtener modelos para el módulo de comprensión del lenguaje natural, el significado de todos los turnos de usuario se etiquetó usando el concepto de *frame* como unidad básica (Segarra et al., 2000). El llamado *frame canónico* constituye un formato estándar de intercambio de información entre los módulos de comprensión y gestor de diálogo. En el *frame canónico* tienen cabida, adecuadamente codificados, los conceptos, atributos y valores de atributos que constituyen el significado de cada turno de usuario.

En los *frames* de consulta (petición de información), los usuarios pueden solicitar los siguientes conceptos: HORA-SALIDA, HORA-LLEGADA, PRECIO, ESTACIÓN-ORIGEN, ESTACIÓN-DESTINO, TIEMPO-RECORRIDO, PARA-EN, TIPO-TREN, SERVICIOS, FECHA-SALIDA y FECHA-LLEGADA. Estos mismos identificadores, puestos entre paréntesis, constituyen las etiquetas de los correspondientes *frames*.

Existen otros *frames* dependientes de la tarea: el *frame* incompleto (cuya etiqueta es una cadena vacía entre paréntesis), usado para etiquetar el aporte de nuevos datos a una consulta expresada en algún turno anterior; el *frame* CONFIRMACIÓN, que etiqueta las peticiones explícitas de confirmación de datos; el *frame* NO-ENTENDIDO, usado para etiquetar las solicitudes de repetición de información o, en general, la falta de comprensión del usuario; y el *frame* de vuelta, caso particular de *frame* de consulta relativo al viaje de vuelta y que se etiqueta añadiendo el sufijo -V a las etiquetas usadas en las consultas sobre el viaje de ida.

Por último, se han considerado tres *frames* independientes de la tarea: el *frame* AFIRMACIÓN, el *frame* NEGACIÓN y el *frame* CIERRE, que etiquetan, respectivamente, las afirmaciones, negaciones y frases de despedida del usuario.

En cualquiera de los *frames* dependientes de la tarea, puede aparecer una lista de nombres y valores de atributos, correspondientes a unidades de información propias de la tarea que hayan recibido valor en la intervención del usuario. La siguiente lista enumera los atributos considerados: CIUDAD-ORIGEN, CIUDAD-DESTINO, PARA-EN, ESTACIÓN-ORIGEN, ESTACIÓN-DESTINO, FECHA-SALIDA, FECHA-LLEGADA, INTERVALO-FECHA-SALIDA, INTERVALO-FECHA-LLEGADA, TIPO-DÍA-SALIDA, TIPO-DÍA-LLEGADA, HORA-SALIDA, HORA-LLEGADA, INTERVALO-HORA-SALIDA, INTERVALO-HORA-LLEGADA, TIPO-TREN, TIPO-VIAJE, CLASE-BILLETE, NÚMERO-RELATIVO-ORDEN, PRECIO, SERVICIOS y TIEMPO-RECORRIDO. Además, para aquellos atributos que no reciben valor en el turno, pero sí en turnos anteriores, y a los que se hace referencia en el turno actual, se definieron las siguientes etiquetas: TREN-ACTUAL, HORA-ACTUAL, FECHA-ACTUAL, MES-ACTUAL, TRAYECTO-ACTUAL, CIUDAD-ORIGEN-ACTUAL y CIUDAD-DESTINO-ACTUAL.

En la Figura 3.1 se muestra un ejemplo de turno de usuario etiquetado mediante *frames*. Se trata de una petición de horarios de ida y de horarios de vuelta, lo que determina que se etiquete con dos conceptos o *frames*. Además, en la misma petición se citan los valores de

los atributos origen, destino, fecha de ida y de vuelta, e intervalo horario, lo que determina la presencia, en la representación semántica, de cinco pares de atributos, según la notación NOMBRE: VALOR.

U0: <i>Hola, quería horarios de Zaragoza a Ciudad Real el día dos de Noviembre y volver el cinco de Noviembre, el dos de Noviembre quiero ir por la mañana, ¿qué es lo que hay?</i>	
U0:	(HORA-SALIDA)
	CIUDAD-ORIGEN: Zaragoza
	CIUDAD-DESTINO: Ciudad-Real
	FECHA-SALIDA: 02-11-????
	INTERVALO-HORA-SALIDA: 05.00-13.00
	(HORA-SALIDA-V)
	FECHA-SALIDA: 05-11-????

Figura 3.1: Ejemplo de representación semántica

3.4 Etiquetado del corpus: actos de diálogo

Para obtener modelos para el gestor del diálogo, se necesita etiquetar todos los turnos de usuario y de sistema. En el caso de los turnos de usuario, aunque existan claras coincidencias con el etiquetado de comprensión, es preciso proceder a un nuevo etiquetado, dado que el uso del mismo será diferente, y también lo será la perspectiva del etiquetado.

Con vistas al módulo de comprensión, cada turno de usuario en el corpus se etiqueta aisladamente, sin considerar el contenido de los otros turnos del diálogo del que forma parte. Tal perspectiva es adecuada en el etiquetado de comprensión puesto que el módulo de comprensión tendrá que abstraer el significado de cada turno de usuario (generar los correspondientes frames canónicos) considerando, exclusivamente, las frases que formen ese turno. En cambio, el gestor de diálogo tendrá que decidir la respuesta del sistema considerando el conjunto del diálogo en curso. Conviene, por tanto, que dicha circunstancia sea tenida en cuenta en el etiquetado a partir del cual se aprenderán los modelos del gestor del diálogo. Así pues, en el etiquetado de diálogo, es habitual resolver las dudas o ambigüedades que plantee el etiquetado de un cierto turno a la vista de los turnos precedentes (en particular, del último) e incluso de los turnos siguientes.

El etiquetado para gestión de diálogo se fundamenta en el concepto de *acto de diálogo*, al igual que en comprensión se basa en el concepto de frame. Se ha considerado una estructura de actos de diálogo constituida por una jerarquía de tres niveles:

- ❑ El primer nivel describe la función genérica de un turno, o segmento de turno, de diálogo (funciones como aperturas y cierres de diálogo, preguntas, respuestas, confirmaciones, etc.) y es independiente de la tarea. Se han definido las siguientes etiquetas para este nivel: APERTURA, CIERRE, INDEFINIDA, NO-ENTENDIDO, ESPERA,

NUEVA-CONSULTA, AFIRMACIÓN, NEGACIÓN, PREGUNTA, CONFIRMACIÓN y RESPUESTA.

- ❑ El segundo nivel, que es específico de la tarea, está relacionado con la representación semántica del turno (es decir, incluye los conceptos que son objeto de pregunta, de confirmación o de respuesta como, por ejemplo, la hora de salida, el precio o el tipo de tren). Las etiquetas definidas para el segundo nivel son las siguientes: HORA-SALIDA, HORA-SALIDA-VUELTA, HORA-LLEGADA, HORA-LLEGADA-VUELTA, PRECIO, ORIGEN, DESTINO, TIEMPO-RECORRIDO, PARA-EN, DÍA-SALIDA, DÍA-LLEGADA, TIPO-TREN, TIPO-VIAJE, SERVICIO, NIL. Esta última es una etiqueta especial para los casos donde el nivel esté vacío.
- ❑ En el tercer nivel se representan los atributos que reciben valores en el turno de diálogo. Las etiquetas definidas para el tercer nivel son las siguientes: HORA-SALIDA, HORA-LLEGADA, PRECIO, ORIGEN, DESTINO, ESTACIÓN-ORIGEN, ESTACIÓN-DESTINO, TIEMPO-RECORRIDO, PARADA-EN, DÍA, TIPO-DÍA, TIPO-TREN, NÚMERO-TRENES, NÚMERO-RELATIVO-ORDEN, TIPO-VIAJE, SERVICIO, NIL.

Como se puede apreciar, las coincidencias con conceptos y atributos etiquetados en comprensión son numerosas.

La notación seguida para identificar un acto de diálogo es la siguiente:

- ❑ Concatenar en el descriptor los identificadores del interlocutor (usuario o mago) y de los tres niveles de etiquetado, separados por el carácter “:”.
- ❑ En los niveles segundo y tercero, si se incluye más de una etiqueta, concatenarlas separadas por el carácter “,”. En el primer nivel del etiquetado sólo puede aparecer una etiqueta.
- ❑ Escribir entre paréntesis el descriptor del acto de diálogo.

Como ejemplo de este etiquetado, en la Figura 3.2 se reproduce un segmento de diálogo con los correspondientes identificadores de actos de diálogo. En el turno U0 del ejemplo, el usuario pregunta (1º nivel de etiquetado) horarios (2º nivel de etiquetado) y aporta los valores de cuatro atributos (los citados en el 3º nivel de etiquetado). En el turno M1, el mago, aparte de reiterar su incompreensión, formula una confirmación de tres atributos (2º nivel de etiquetado) y aporta los valores de los tres atributos (razón de que se repitan las etiquetas en el 3º nivel). En el turno U1, el usuario responde (1º nivel) con el atributo fecha (2º nivel) dando su valor (3º nivel). En el turno M2, el mago confirma el objetivo de la consulta, los horarios (2º nivel) y repite los valores de tres atributos (3º nivel).

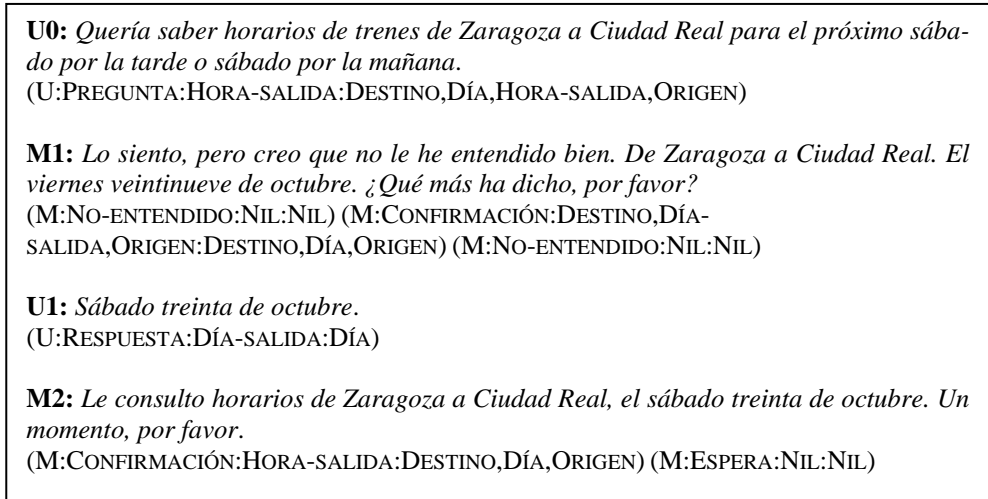


Figura 3.2: Ejemplo de etiquetado de actos de diálogo

En lo sucesivo, tanto los descriptores de actos de diálogo del sistema como los turnos (frases o frames) correspondientes al sistema se marcarán con una **S** (de sistema) en lugar de la **M** (de mago) usada en esta figura. Igualmente, el identificador DÍA-SALIDA será sustituido por el identificador FECHA-SALIDA.

4. Módulo de comprensión

4. Módulo de comprensión

4.1 Introducción

En esta sección se describe el módulo de comprensión del lenguaje natural desarrollado en la presente tesis. Este módulo recibe una entrada de texto, procedente del reconocedor del habla, y genera una salida adecuada, para el módulo gestor de diálogo, consistente en la representación semántica de las frases recibidas.

Tal como se indicó en la Sección 1, el principal objetivo de esta tesis es el desarrollo del módulo gestor de diálogo. El desarrollo de un módulo de comprensión se planteó como un objetivo adicional, o secundario, puesto que en el grupo de investigación de “Reconocimiento de Formas e Inteligencia Artificial” (DSIC-UPV) ya se disponía de un módulo de comprensión basado en modelos estocásticos para la tarea BASURDE (Segarra et al., 2002).

En (Segarra et al., 2002), la comprensión del lenguaje se trata como una tarea de traducción que consta de dos fases. En la primera fase, el texto de entrada se traduce a representaciones semánticas, definidas en un lenguaje semántico intermedio (ISL) secuencial, y en la segunda fase, dichas representaciones semánticas se traducen a frames. En la traducción a ISL se aplican técnicas de aprendizaje automático (transductores secuenciales), y en la traducción a frames se aplica un sencillo sistema basado en reglas.

En este nuevo módulo de comprensión, la tarea se aborda también en dos fases. En la primera fase se identifican los atributos, mediante la consulta de diccionarios que almacenan los posibles valores de los atributos, y se reescribe la frase sustituyendo valores de atributos por identificadores de variables atributo. Esta reescritura permite, en la segunda fase de la comprensión, identificar y generar los frames mediante técnicas estadísticas. Se han estimado modelos estocásticos de comprensión que establecen las probabilidades de que las secuencias de palabras recibidas sean muestras de determinados actos de diálogo. Y se ha considerado la estrecha vinculación entre la representación semántica mediante frames, usada en comprensión, y el etiquetado de actos de diálogo, usado en la gestión del diálogo.

En esta sección, se describe dicho módulo de comprensión, se presentan los resultados de su evaluación y se compara con el funcionamiento del anterior módulo (Segarra et al., 2002), concluyendo con una valoración del módulo desarrollado.

4.2 Descripción del módulo de comprensión

Tal como se ha indicado, el corpus BASURDE fue etiquetado para estimar modelos estocásticos de comprensión y diálogo. En comprensión del lenguaje, se usaron frames como unidades semánticas que almacenan la información relevante contenida en un turno de usuario. Un frame consta de un nombre de concepto y, opcionalmente, uno o más pares {nombre, valor} de atributos. Un ejemplo de etiquetado de dos turnos consecutivos de usuario se muestra en la Figura 4.1. En el etiquetado de diálogo, a cada turno del corpus se le asignaron los identificadores de uno o varios actos de diálogo. En la Figura 4.2 se presenta un fragmento de diálogo etiquetado en términos de actos de diálogo. Los turnos de usuario son los mismos de la Figura 4.1.

U0: *Quería saber horarios de trenes de Zaragoza a Ciudad Real para el próximo sábado por la tarde.*
 (HORA-SALIDA)
 ORIGEN: Zaragoza
 DESTINO: Ciudad-Real
 FECHA-SALIDA: 09-06-2001
 INTERVALO-HORA-SALIDA: 13.00-21.00
U1: *Sábado treinta de octubre.*
 ()
 FECHA-SALIDA: 30-10-????

Figura 4.1: Etiquetado de comprensión

U0: *Quería saber horarios de trenes de Zaragoza a Ciudad Real para el próximo sábado por la tarde.*
 (U:PREGUNTA:HORA-SALIDA:DESTINO,FECHA-SALIDA,HORA-SALIDA,ORIGEN)
S1: *Lo siento, pero creo que no le he entendido bien. De Zaragoza a Ciudad Real. El viernes veintinueve de octubre. ¿Qué más ha dicho, por favor?*
 (S:NO-ENTENDIDO:NIL:NIL) (S:CONFIRMACIÓN:DESTINO,FECHA-SALIDA,ORIGEN:DESTINO,FECHA-SALIDA,ORIGEN) (S:NO-ENTENDIDO:NIL:NIL)
U1: *Sábado treinta de octubre.*
 (U:RESPUESTA:FECHA-SALIDA:FECHA-SALIDA)
S2: *Le consulto horarios de Zaragoza a Ciudad Real, el sábado treinta de octubre. Un momento, por favor.*
 (S:CONFIRMACIÓN:HORA-SALIDA:DESTINO,FECHA-SALIDA,ORIGEN)
 (S:ESPERA:NIL:NIL)

Figura 4.2: Ejemplo de etiquetado de actos de diálogo

Se observa que hay una estrecha relación entre los etiquetados de diálogo y de comprensión. Este hecho justifica la aproximación llevada a cabo en el desarrollo del módulo de comprensión que en esta sección se presenta.

En la Tabla 4.1 se muestran las correspondencias entre los frames y los actos de diálogo más frecuentes.

<i>etiqueta de frame</i>	<i>etiqueta de acto de diálogo</i> <i>Nivel1:Nivel2</i>
(HORA-SALIDA)	PREGUNTA:HORA-SALIDA
(HORA-SALIDA-VUELTA)	PREGUNTA:HORA-SALIDA-VUELTA
(HORA-LLEGADA)	PREGUNTA:HORA-LLEGADA
(PRECIO)	PREGUNTA:PRECIO
(SERVICIOS)	PREGUNTA:SERVICIOS
(TIPO-TREN)	PREGUNTA:TIPO-TREN
(NO-ENTENDIDO)	NO-ENTENDIDO:NIL
(AFIRMACIÓN)	AFIRMACIÓN:LISTA-ATRIBUTOS
(NEGACIÓN)	NEGACIÓN: LISTA-ATRIBUTOS
()	RESPUESTA: LISTA-ATRIBUTOS
(CIERRE)	CIERRE:NIL

Tabla 4.1: Frames y actos de diálogo

Parece razonable aprender las secuencias de palabras que aparecen en los turnos asignados a cada clase de acto de diálogo y sus frecuencias. Estos modelos de secuencias de palabras, incorporados al módulo de comprensión, permitirán predecir los frames que, con más probabilidad, representen el significado de los turnos de usuario.

El aprendizaje de estos modelos, sin embargo, llevaría a tener un número elevado de distintas secuencias de palabras con frecuencias de aparición bajas, debido a la reducida extensión del corpus (14.902 palabras) en relación al tamaño de su vocabulario (637 palabras). Por ello, antes de aprender los modelos, fue necesario realizar un preproceso del corpus, que consistió en una categorización de las instancias de los atributos. Mediante este preproceso, una frase como, por ejemplo, “Quería ir a Valencia el miércoles por la noche” se convierte en “Quería ir a <DESTINO> el <DIA> por la <INTERVALO-HORA-SALIDA>”. Hemos aprendido los modelos de secuencias de palabras a partir del corpus preprocesado, que contiene menos secuencias distintas de palabras y con frecuencias más altas que el corpus original.

Las frases del corpus preprocesado se han clasificado en 11 clases que corresponden a los frames mostrados en la Tabla 4.1. Para cada clase, se ha obtenido las frecuencias de las secuencias de palabras de longitud en el rango $n \in [1, 6]$ y frecuencia mayor a 1. Se ha construido un modelo para cada valor n .

Cada modelo se representa mediante un matriz de frecuencias. Sea mf_n , la matriz de un modelo de secuencias de n palabras. Es una matriz de R filas y C columnas, donde R es el número de secuencias, distintas y no únicas (con frecuencia mayor a 1) de n palabras y C es el número de clases. R es igual a 369, 1174, 1360, 1087, 758 y 476 en las matrices mf_1 , mf_2 , mf_3 , mf_4 , mf_5 y mf_6 , respectivamente; C es igual a 11 en todas las matrices. El número almacenado en una posición $[j, k]$ de una matriz, mf_n $[j, k]$, representa la frecuencia de la secuencia de n palabras número j en la clase número k .

```

// 1ª fase: identificación de los atributos y reescritura de las frases

    // para cada palabra  $w_n$  de la frase,  $S$ , y cada atributo  $a_t$  de la tarea
FOR (n=1..|S|; t = 1..A) DO
    // buscar las ocurrencias de los valores de todos los atributos
    IF  $w_n$  = instancia de  $a_t$ 
        // asignar la palabra  $w_n$  al atributo  $a_t$  de la lista de atributos
        lista-atributos = asignar( $a_t$ ,  $w_n$ );
        // en  $S$ , sustituir la palabra  $w_n$  con su etiqueta de atributo
        S = sustituir( $w_n$ , etiqueta-atributo( $a_t$ ));
    END-IF;
END-FOR;

// 2ª fase: identificación y generación de los frames

P =  $\emptyset$ ; // P = prefijo de frase, inicial cadena vacía
n = 1; // n = cursor inicial en 1ª palabra,  $w_1$ 
m = 1; // m = n° de siguiente frame a generar;
N = 6; C = 11; // N = n° de modelos; C = n° de clases
WHILE n ≤ |S| // mientras el prefijo no sea igual a la frase, S

    P = P  $w_n$  // [a] concatenar siguiente palabra al prefijo

    // [b] construir listas de las secuencias de palabras en prefijo
 $\forall i / 1 \leq i \leq \min(n, N) : L_i = \text{list}(P)$ 

    // [c] calcular la matriz de acumulación de frecuencias
 $\forall i / 1 \leq i \leq N, \forall k / 1 \leq k \leq C : \text{maf}[i, k] = \sum_{(j = 1..|\text{mf}_i|) \wedge (j \in L_i)} \text{mf}_i[j, k]$ 

    // [d] clasificar el prefijo en un acto de diálogo
 $\forall k / 1 \leq k \leq C :$ 
        
$$p[k] = \frac{\sum_{(i = 1..N)} (N+1-i) \cdot \text{maf}[i, k]}{\sum_{(i = 1..N, k = 1..C)} (N+1-i) \cdot \text{maf}[i, k]}$$


    // [e] seleccionar frame candidato k con probabilidad  $p[k]$ 
    candidato-id[n] =  $\text{argmax}_k p[k]$ ;
    candidato-prob[n] =  $\max_k p[k]$ ;

    r = n-1; // [f] opcionalmente, generar frame k con probabilidad  $p[k]$ 
    IF (candidato-id[n] = candidato-id[r]) & (candidato-prob[n] > umbral)
        frame[m] = candidato[n];
        WHILE (candidato-id[n] = candidato-id[r]) r = r-1;
            // completar frame con atributos
        frame-atributos[m] = seleccionar(lista-atributos, r+1, n);
        m = m+1; // avanzar el contador de frames
    END-IF;
    n = n+1; // avanzar el cursor de la frase
END-WHILE;

```

Figura 4.3: Algoritmo del módulo de comprensión

El módulo de comprensión traduce las frases del usuario identificando, primero, los atributos y, después, los frames. El algoritmo se muestra en la Figura 4.3. En la identificación de los atributos, se reescribe la frase del mismo modo que se realizó en el preproceso del corpus y se construye una lista de instancias de atributos para asignar después a los frames.

La identificación de los frames consta de los siguientes pasos (indicados como [a], [b]..., en los comentarios del algoritmo de la Figura 4.3):

- ❑ Concatenar la siguiente palabra al prefijo de la frase (inicialmente, el prefijo es igual a la cadena vacía, y el cursor de la frase apunta a la primera palabra).
- ❑ Listar las secuencias de n palabras que haya en el prefijo.
- ❑ A partir de las matrices modelo, mf_i , y de la lista L_i , calcular una matriz que llamamos *matriz de acumulación de frecuencias*, maf .
- ❑ A partir de maf , calcular las probabilidades $p[k]$ y clasificar el prefijo en una clase de acto de diálogo.
- ❑ Seleccionar un frame candidato.
- ❑ Generar un frame con cierta probabilidad.
- ❑ Repetir hasta que el prefijo sea igual a la frase.

La matriz de acumulación de frecuencias, maf , se obtiene a partir de las matrices mf_i de secuencias de n palabras (los modelos) y de la lista L_i de secuencias de n palabras halladas en el segmento de la frase, según la siguiente fórmula:

$$\forall i / 1 \leq i \leq N, \forall k / 1 \leq k \leq C : maf [i, k] = \sum_{(j = 1..|mf_i|) \wedge (j \in L_i)} mf_i [j, k]$$

La matriz se calcula cada vez que una palabra se añade al prefijo. Así, cuando el prefijo contiene las primeras q palabras de la frase ($P = w_1..w_q$), el número almacenado en una posición $[i, k]$ de maf representa la suma de las frecuencias, almacenadas en la clase número k del modelo de secuencias de i palabras, para las secuencias de i palabras que aparecen en el prefijo $w_1..w_q$.

La clasificación del prefijo como instancia de un acto de diálogo se realiza mediante el cálculo de los vectores p . Consideramos que $p[k]$ representa la probabilidad de que el prefijo corresponda al acto de diálogo número k . Dada la correspondencia entre frames y actos de diálogo, esto determina el frame candidato.

Un candidato k se convierte en un frame k cuando se cumplen dos condiciones:

- ❑ Su $p[k]$ debe ser superior a cierto umbral experimental (indicado en el algoritmo como condición: $candidato-prob[n] > umbral$).
- ❑ El candidato debe ser el mismo al menos durante dos palabras consecutivas (indicado en el algoritmo como condición: $candidato-id[n] = candidato-id[r]$).

En tal caso, se genera un frame con el identificador y probabilidad del candidato. Este frame se completa con los atributos hallados en el segmento donde el frame generado tiene el máximo local $p[k]$.

4.3 Incorporación de otras fuentes de información

A partir de una evaluación preliminar y del estudio de los casos de clasificaciones erróneas, hemos diseñado algunas fuentes de información complementarias (palabras clave y reglas) y las hemos incorporado al módulo de comprensión.

Hemos catalogado 25 palabras como palabras clave de actos de diálogo. Algunos ejemplos son: “llegada” es una palabra clave del frame (HORA-LLEGADA); “sí” es una palabra clave del frame (AFIRMACIÓN).

Cuando se encuentra una palabra clave, se activa un indicador asociado al acto de diálogo correspondiente. Más tarde, cuando el módulo de comprensión decide cuáles frames candidatos serán generados y cuáles no, aplica diferentes umbrales dependiendo de los indicadores activos (el umbral de un frame k se reduce si está activo el indicador k).

Además, hemos definido varias reglas para resolver algunos problemas, usando los indicadores de palabras clave como pistas de cuáles frames han de generarse. Los conjuntos de reglas considerados son los siguientes:

- ❑ Conjunto de reglas 1. Reglas de reasignación de los frames de horarios:
 - Regla 1a. Si el frame candidato es (HORA-SALIDA) pero el indicador de (HORA-LLEGADA) está activado, entonces se genera un frame (HORA-LLEGADA) en lugar del frame candidato.
 - Regla 1b. Si el frame candidato es (HORA-SALIDA) pero el indicador de (HORA-SALIDA-VUELTA) está activado, entonces se genera un frame (HORA-SALIDA-VUELTA) en lugar del frame candidato.
- ❑ Conjunto de reglas 2. Reglas de detección de falsos frames de cierre:
 - Regla 2a. Si (CIERRE) es frame candidato pero no es único (le siguen otros candidatos), entonces se genera un frame (NEGACIÓN) en lugar del frame candidato (CIERRE).
 - Regla 2b. Si el 1º frame candidato es (NEGACIÓN) y el 2º frame candidato es (CIERRE), entonces se genera sólo el frame (NEGACIÓN).
- ❑ Conjunto de reglas 3. Reglas de asignación correcta de los atributos intervalo de fechas. En el etiquetado de BASURDE, dos valores de fechas se concatenan como valor del atributo FECHA-SALIDA, en lugar del atributo INTERVALO-FECHA-SALIDA. Este conjunto de reglas realiza tal cambio para satisfacer el etiquetado de referencia.

- ❑ Conjunto de reglas 4. Si no se genera ningún frame pero hay indicadores activos, entonces se genera un frame por cada indicador activo. Si ningún indicador está activo, entonces se genera un frame incompleto, etiquetado como ().
- ❑ Conjunto de reglas 5. Si hay atributos que no se han asignado a ningún frame y hay indicadores activos, entonces se genera un frame por cada indicador activo. Si ningún indicador está activo, entonces se genera un frame incompleto.
- ❑ Conjunto de reglas 6. Si sólo hay un candidato de frame, y es (AFIRMACIÓN) o (NEGACIÓN), y hay indicadores activos, entonces se genera un frame por cada indicador activo.

4.4 Evaluación

Para evaluar el funcionamiento del módulo de comprensión, se le ha suministrado las frases de usuario del corpus de BASURDE y se han comparado los frames generados por el módulo de comprensión y los frames del etiquetado de referencia del corpus. Hemos considerado la medida llamada *tasa de unidades de frame correctas* (*corrected frame units rate*, %cfu), que es el porcentaje de nombres de frame correctos y de pares {nombre, valor} de atributos correctos, calculado mediante:

$$\%cfu = (1 - (nI + nS + nB) / (nS + nB + nC)) * 100$$

donde:

nI = número de inserciones

nS = número de sustituciones

nB = número de borrados

nC = número de unidades correctas

En la Tabla 4.2 se detallan los resultados obtenidos. Tal como se ha descrito, el módulo de comprensión sigue procedimientos bien distintos para identificar los nombres de frames y los valores de los atributos. Por ello, además de la tasa de unidades de frame correctas (%cfu), en la tabla se informa de su desglose en *tasa de nombres de frame correctos* (%cf) y *tasa de atributos correctos* (%ca). Por otra parte, se ha medido la contribución de los distintos modelos a las tasas de acierto, así como los efectos de incorporar las palabras clave y las reglas.

Modelos	- reglas			+ reglas		
	% cf	% ca	% cfu	% cf	% ca	% cfu
mf _i , 1 ≤ i ≤ 1	69,35	60,30	64,47	81,30	78,50	79,79
mf _i , 1 ≤ i ≤ 2	70,07	63,64	66,61	80,14	78,17	79,08
mf _i , 1 ≤ i ≤ 3	71,04	64,59	67,56	80,39	78,13	79,17
mf _i , 1 ≤ i ≤ 4	71,18	64,30	67,47	80,43	77,55	78,88
mf _i , 1 ≤ i ≤ 5	71,42	64,26	67,56	80,67	77,71	79,08
mf _i , 1 ≤ i ≤ 6	71,90	64,22	67,76	81,64	78,62	80,01
mf _i , 1 ≤ i ≤ 6 + palabras clave	72,29	62,32	66,92	84,53	79,41	81,77

Tabla 4.2. Resultados de la evaluación

Se puede observar que la inclusión de las reglas y las palabras clave proporciona una mejora significativa: de 12,63% en *tasa de nombres de frame correctos* (%cf = 71,90% si se usan sólo los 6 modelos de secuencias de palabras; %cf = 84,53% si, además, se usan reglas y palabras clave), y de 14,01% en *tasa de unidades de frame correctas* (%cfu = 67,76% si se usan sólo los 6 modelos de secuencias de palabras; %cfu = 81,77% si, además, se usan reglas y palabras clave).

Hemos estudiado las contribuciones particulares de los diferentes conjuntos de reglas. En la Tabla 4.3 se resumen los efectos de las reglas sobre los resultados.

	% cf	% ca	% cfu
modelos + palabras clave	72,29	62,32	66,92
modelos + palabras clave + conjunto de reglas nº 1	72,34	62,94	67,27
modelos + palabras clave + conjunto de reglas nº 2	72,24	62,32	66,90
modelos + palabras clave + conjunto de reglas nº 3	72,29	63,43	67,52
modelos + palabras clave + conjunto de reglas nº 4	76,96	67,89	72,08
modelos + palabras clave + conjunto de reglas nº 5	84,00	77,67	80,59
modelos + palabras clave + conjunto de reglas nº 6	74,51	62,32	67,94
modelos + palabras clave + todas las reglas	84,53	79,41	81,77

Tabla 4.3. Efectos de las reglas

Del análisis de la Tabla 4.3, se concluye que existe un amplio solapamiento entre los diferentes conjuntos de reglas. Las reglas que añaden frames conforme a los indicadores activos, cuando los modelos de secuencias de palabras no generan frames (conjunto de reglas nº 4) o cuando quedan atributos pendientes de asignar a frames (conjunto de reglas nº 5) son las que proporcionan las mejoras más apreciables. Por otra parte, la inclusión de las reglas que detectan los falsos cierres (conjunto de reglas nº 2) causa una pequeña degradación en %cf. Sin embargo, hemos mantenido este conjunto de reglas dada su importancia para evitar la conclusión prematura de los diálogos cuando el módulo de comprensión está integrado en el sistema de diálogo.

4.5 Comparativa y conclusiones

Los resultados más recientes de evaluación del módulo de comprensión de (Segarra et al., 2002) pueden encontrarse en (Hurtado et al., 2004). La tabla 4.4 resume los datos relativos al porcentaje de aciertos en unidades de frames.

Modelos	% cfu
Bi-multigram – BI	85,7
Bi-multigram – TRI	86,1
Bi-multigram – PTA	86,0
Bi-multigram – MGGI	86,3
Bi-multigram – ECGI	87,2

Tabla 4.4. Resultados de Hurtado et al., 2004

El mejor resultado obtenido con el módulo de comprensión presentado en esta sección es 81,77%. Por tanto, el módulo propuesto tiene un comportamiento en torno a un 5% peor que las versiones más recientes del anterior módulo.

Sin embargo, el funcionamiento del módulo propuesto es aceptable. Era previsible que no alcanzara las prestaciones del anterior módulo, dado que utiliza bastante menos información estadística (sólo las frecuencias de las palabras o clases de palabras) y que no se vale de la traducción a un lenguaje semántico intermedio. Esta sencillez, y el hecho de haber sido programado en lenguaje C (en vez de Python, lenguaje usado en el anterior módulo), explica su principal ventaja: su velocidad al realizar la traducción a frames, muy superior a la del anterior módulo.

Esta propiedad, su velocidad, resulta decisiva para justificar la conveniencia del desarrollo de este módulo de comprensión que, en todo momento, tuvo un carácter auxiliar en el desarrollo del sistema de diálogo. El objetivo principal de la tesis es el desarrollo y evaluación del gestor de diálogo. Con este módulo de comprensión, es posible realizar un número mucho mayor de pruebas del gestor de diálogo.

La ventaja aún se vuelve más importante en el momento de realizar experimentos con simulador de usuario, donde se programan series de varios miles de diálogos que pueden requerir varios días de ejecución. En estos experimentos, el uso del módulo de comprensión propuesto supone multiplicar, al menos, por 3 la velocidad de ejecución de los diálogos simulados.

Además, las frases del simulador de usuario son traducidas correctamente por ambos módulos de comprensión, de modo que, en esas condiciones experimentales, resulta preferible en todos los sentidos usar este nuevo módulo de comprensión del lenguaje.

5. Gestor de diálogo

5. Gestor de diálogo

5.1 Introducción

En esta sección se presenta la parte principal del sistema desarrollado en esta tesis, el módulo gestor de diálogo. Información sobre el trabajo de investigación llevado a cabo en este módulo gestor de diálogo puede encontrarse en (Torres et al., 2002a), (Torres et al., 2002b), (Torres et al., 2003), (Torres et al., 2004) y (Torres et al., 2005a).

En primer lugar, se describe la entrada al gestor de diálogo y su interconexión con los otros módulos del sistema de diálogo. A continuación, se expone el procedimiento seguido en el aprendizaje de los modelos estocásticos de diálogo y las características de los modelos obtenidos.

Después, en el apartado 5.4, se presenta el algoritmo del gestor de diálogo y, en el apartado 5.5, se describe el registro histórico, fuente de información complementaria de los modelos estocásticos de diálogo. A continuación, los apartados siguientes se dedican a exponer en detalle varios aspectos relevantes del funcionamiento del gestor de diálogo:

- ❑ El uso de las técnicas de generalización semántica (o suavizado de los frames de usuario) en la selección de transiciones a estados de usuario.
- ❑ El uso de las reglas de consistencia con el contenido del registro histórico (o uso de la historia del diálogo como complemento a los modelos de bigramas) en la selección de transiciones a estados de sistema.
- ❑ El uso de las medidas de confianza y su efecto en la selección de transiciones en el modelo estocástico.

Finalmente, en el apartado 5.9, se ilustra el funcionamiento del gestor de diálogo mediante un ejemplo de diálogo, expuesto con bastante detalle.

5.2 Entrada al gestor de diálogo

En el sistema de diálogo, el módulo gestor de diálogo se integra con tres módulos: un módulo de comprensión, un módulo de consulta a la base de datos y un módulo generador de respuestas. Estos módulos se interconectan del modo indicado en la Figura 5.1.

El módulo de comprensión traduce las frases en lenguaje natural del usuario a una representación semántica (en nuestro sistema, a un conjunto de frames que representan el significado de cada turno de usuario). Las entradas al gestor de diálogo son estos frames o representaciones semánticas de los turnos de usuario. En la Figura 5.2 se muestra un ejemplo de turno de usuario y su representación semántica, en términos de frame para la tarea BASURDE. Cuando el gestor de diálogo ha decidido cómo responderá el sistema al usuario, genera otro conjunto de frames. Estos frames se suministran al siguiente módulo, el generador de respuestas, que realiza la traducción inversa: de una representación semántica del sistema a una respuesta del sistema en lenguaje natural.

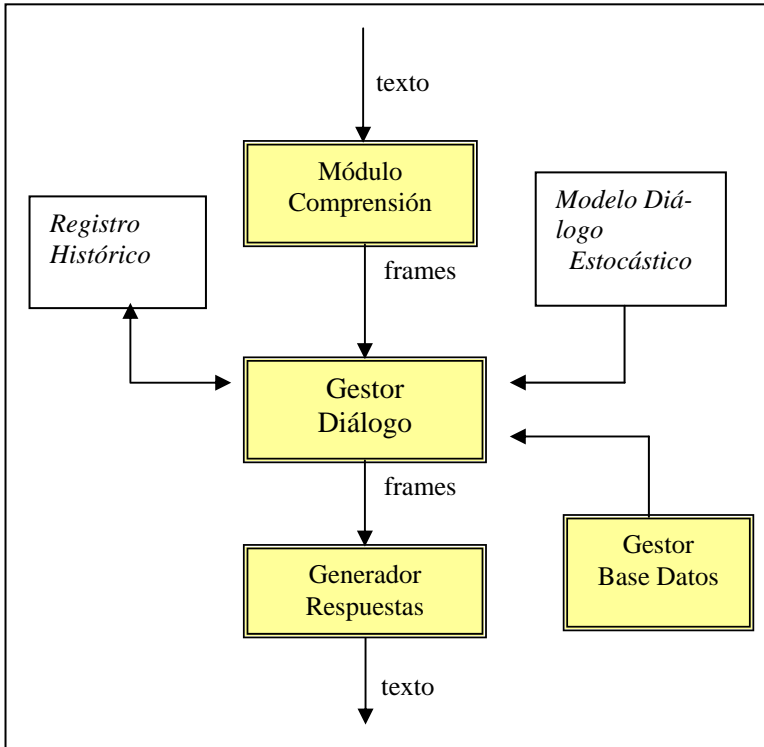


Figura 5.1: Interconexión del gestor de diálogo

U0: *Quería los horarios de trenes para ir de Burgos a Bilbao el dos de marzo por la mañana.*
 (HORA-SALIDA) * [0.85, 0.92]
 ORIGEN: Burgos * [0.82, 0.85]
 DESTINO: Bilbao * [0.82, 0.85]
 FECHA-SALIDA: 02-03-2005 * [0.73, 0.80]
 INTERVALO-HORA-SALIDA: 05.00-13.00 * [0.71, 0.62]

Figura 5.2: Ejemplo de representación semántica

Esta vez, en la representación semántica de la Figura 5.2, se han indicado las medidas de confianza asociadas a los conceptos y atributos, medidas procedentes del módulo de comprensión (García et al., 2003). Cada línea de la representación semántica (sea un identificador de frame, sea un par nombre-valor de atributo) aparece seguida de un par de valores en el formato “* [d.dd, d.dd]”, donde *d.dd* indica un número real entre 0 y 1 con precisión de dos decimales. La primera medida de confianza corresponde a una combinación lineal de la confianza de la palabra, dada por el reconocedor del habla, y de la probabilidad del unigrama de la palabra, en el segmento asociado a una unidad semántica. La segunda medida de confianza corresponde a la probabilidad de la unidad semántica en la secuencia de unidades semánticas, según el modelo de lenguaje semántico.

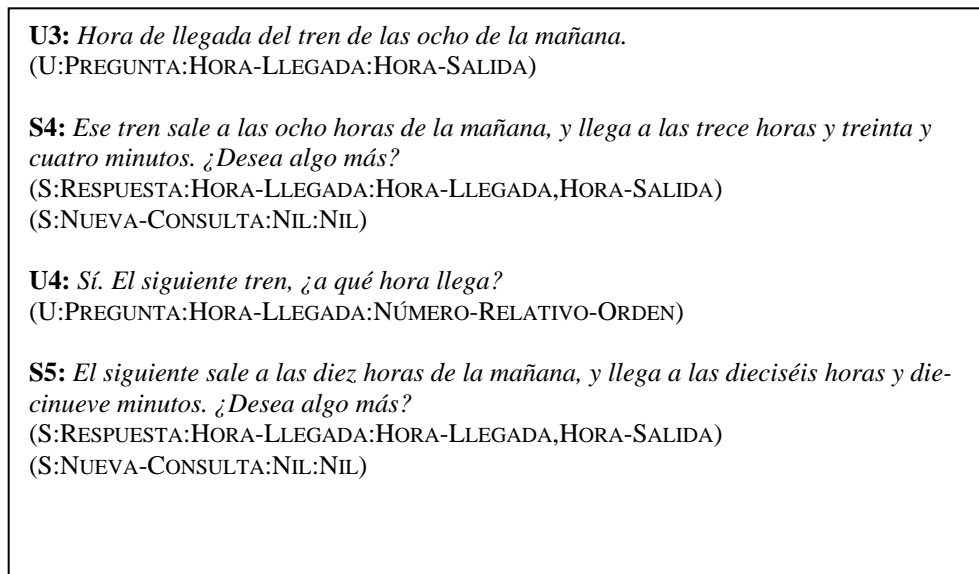


Figura 5.3: Ejemplo de etiquetado de actos de diálogo

5.3 Aprendizaje de los modelos de diálogo

Las estructuras de los modelos de diálogo se representan por medio de secuencias de actos de diálogo. Como se ha expuesto anteriormente, se definió un etiquetado de tres niveles, donde el primer nivel describe los actos generales del diálogo, independientemente de la tarea, el segundo nivel incluye los conceptos específicos de la tarea y el tercer nivel incluye los atributos dados en el turno. Cada turno del corpus se etiquetó con los identificadores de uno o varios actos de diálogo de acuerdo con esta propuesta.

Un fragmento de diálogo etiquetado de este modo ya se ha mostrado en la Figura 3.2. Consideremos aquí otro ejemplo, otro segmento de diálogo etiquetado con los tres niveles tal como se muestra en la Figura 5.3.

A partir del corpus de diálogos de BASURDE, se han aprendido modelos de estados finitos estocásticos de diálogo. Cada estado, en estos modelos, corresponde a un cierto acto de diálogo (tiene asociado un determinado descriptor, determinado por el etiquetado del corpus), y existen estados del sistema, asociados a actos de diálogo del sistema, y estados de usuario, asociados a actos de diálogo del usuario. Los modelos considerados son modelos de bigramas. Así pues, dado un cierto estado de diálogo, el modelo determina las transiciones posibles a siguientes estados de diálogo así como las probabilidades de tales transiciones.

Si se considera un modelo que incluya toda la información etiquetada en el corpus, los estados del modelo tendrán asociados, como identificadores o descriptores, las mismas cadenas obtenidas en el etiquetado. Así, en el fragmento de la Figura 5.3, se encuentran 3 estados, los siguientes:

- ❑ *Estado1*: (U:PREGUNTA:HORA-LLEGADA:HORA-SALIDA)
- ❑ *Estado2*: (S:RESPUESTA:HORA-LLEGADA:HORA-LLEGADA,HORA-SALIDA)
- ❑ (S:NUEVA-CONSULTA:NIL:NIL)
- ❑ *Estado3*: (U:PREGUNTA:HORA-LLEGADA:NÚMERO-RELATIVO-ORDEN)

Y las siguientes 3 transiciones:

- ❑ *Estado1* → *Estado2*
- ❑ *Estado2* → *Estado3*
- ❑ *Estado3* → *Estado2*

En el fragmento de la Figura 5.3, se registran dos instancias de un mismo estado de sistema y sólo una instancia de los estados de usuario y, en consecuencia, de las transiciones desde los mismos. Esta situación es algo habitual, dado que en los estados de usuario se refleja la variabilidad de comportamiento de los informantes en la adquisición del corpus, mientras que en los estados del sistema se refleja el comportamiento más homogéneo del sistema, que en el corpus considerado es el Mago de Oz. Si esto se extiende a todo el corpus (al reducido conjunto de los 227 diálogos del corpus), se obtendría un modelo con un número muy elevado de estados, la mayoría con frecuencias de aparición muy bajas. Igualmente, se tendría gran número de transiciones a las que, en general, corresponderían probabilidades muy bajas.

En el ejemplo, se observa que los dos estados de usuario (*Estado1* y *Estado3*) son muy parecidos, difiriendo únicamente en el tercer nivel de etiquetado. Ambos estados corresponden a preguntas sobre horarios de llegada, en un caso con referencia a un horario de salida, en el otro caso con una referencia relativa a número de orden del tren.

Si se omiten estas referencias a valores concretos de atributos (es decir, si se ignora el tercer nivel de etiquetado), ambos estados de usuario se pueden unificar en un mismo estado genérico de consulta de horarios de llegada. Si se aplica el mismo criterio a los estados de sistema, en el ejemplo anterior se encuentran sólo 2 estados, los siguientes:

- ❑ *Estado1*: (U:PREGUNTA:HORA-LLEGADA)
- ❑ *Estado2*: (S:RESPUESTA:HORA-LLEGADA) (S:NUEVA-CONSULTA:NIL)

Y las siguientes 2 transiciones:

- ❑ *Estado1* → *Estado2*
- ❑ *Estado2* → *Estado1*

En el fragmento, entonces, aparecen menos estados y transiciones, pero con frecuencias de aparición mayores. Extendido a todo el corpus, se obtiene un modelo con un número bastante menor de estados, donde las frecuencias de las transiciones son más elevadas.

De este modo, a costa de perder detalle en la descripción de los estados de diálogo, se obtienen modelos donde cada estado se estima a partir de un conjunto más significativo de muestras. En el gestor de diálogo, la información de tercer nivel, no contemplada en el

modelo de diálogo, será almacenada en una estructura complementaria, llamada *registro histórico*, tal como se verá más adelante.

En la Tabla 5.1 se muestran estadísticas relativas a los modelos estocásticos de diálogo que se infieren a partir del corpus de BASURDE según el nivel de etiquetado que se incorpore a dichos modelos.

<i>Modelo</i>	<i>Completo</i>	<i>Específico</i>	<i>Genérico</i>
Niveles de etiquetado incluidos	1°, 2°, 3°	1°, 2°	1°
Nº de muestras (= turnos)	3.528	3.528	3.528
Nº de estados	776	313	74
Nº de estados con 3 o menos muestras	646	222	38
Nº de estados con sólo 1 muestra	496	161	29
Nº medio de muestras por estado	4,55	11,27	47,68
Nº de transiciones	1.962	990	284
Nº de transiciones con sólo 1 muestra	1.635	690	152
Nº medio de muestras por transición	1,80	3,56	12,42

Tabla 5.1. Estadísticas de modelos de diálogo en función del nivel de etiquetado

Se aprecia, en la tabla, que el modelo llamado *Completo* presenta una excesiva dispersión del conocimiento o información relativa a la tarea, con multitud de estados estimados, en promedio, con un número muy escaso de muestras. Así pues, y debido a las limitaciones derivadas del tamaño del corpus de BASURDE, este modelo completo sería claramente inadecuado para establecer una estrategia de diálogo basada en criterios estadísticos.

Por tanto, se han aprendido los modelos estocásticos de diálogo a partir, solamente, de los niveles primero y segundo del etiquetado. Igualmente, dado el reducido tamaño de este corpus de entrenamiento, se han obtenido dos modelos, llamados *Específico* y *Genérico* en la anterior tabla. Cada estado en el modelo específico se identifica mediante una concatenación de una o más cadenas (*Turno:PrimerNivel:SegundoNivel*), donde *Turno* identifica al usuario o al sistema, *PrimerNivel* identifica el acto de diálogo y *SegundoNivel* identifica los conceptos y atributos incluidos en ese acto de diálogo. Los estados en el modelo genérico se identifican mediante una concatenación de una o más cadenas (*Turno:PrimerNivel*) y, en consecuencia, este modelo representa un nivel de conocimiento más general. Como se ha apuntado antes, ambos modelos son modelos de bigramas.

Como se ha indicado, al inferir los modelos automáticamente a partir del corpus, se obtiene un modelo específico con 313 estados y un modelo genérico con 74 estados. Sin embargo, 222 estados del modelo específico y 38 estados del modelo genérico aparecen como mucho tres veces en todo el corpus (y, además, la mayoría de estos estados se corresponden con una sola muestra). Ante estos estados con frecuencias tan bajas se puede considerar que: a) no son representativos de la tarea; b) son representativos, aunque mal representados en el corpus. En el primer supuesto, se puede optar por, directamente, suprimirlos del modelo. En el segundo supuesto, se puede considerar: a) asimilarlos a otros estados de semántica lo más próxima posible y con frecuencias de aparición significativas; b) agrupar varios estados de frecuencias muy bajas, que sean muy similares en su significado, en un único estado. En

ambos casos, el estado que asimila a otros estados, aumenta su masa de probabilidad en concordancia con la que tenían los estados asimilados.

Este proceso de agrupamiento de los estados, en ambos modelos, se ha realizado manualmente. Los resultados se resumen en la Tabla 5.2.

<i>Modelo</i>	<i>Específico</i>		<i>Genérico</i>	
	Inicial	Procesado	Inicial	Procesado
Nº de muestras (= turnos)	3.528	3.404	3.528	3.404
Nº de estados	313	155	74	33
Nº de estados con ≤ 3 muestras	222	64	38	2
Nº de estados con sólo 1 muestra	161	19	29	0
Nº medio de muestras por estado	11,27	21,96	47,68	103,15
Nº de transiciones	990	691	284	156
Nº de transiciones con sólo 1 muestra	690	360	152	46
Nº medio de muestras por transición	3,56	4,93	12,42	21,82

Tabla 5.2. Modelos de diálogo antes y después del agrupamiento de estados

Hay que precisar que en los modelos así procesados se han descartado 124 muestras (turnos de diálogo) correspondientes a situaciones de diálogo consideradas como no representativas de la tarea o que no se ha juzgado conveniente asimilar a ninguno de los estados de los nuevos modelos. Por esta razón, mientras los modelos iniciales se estimaban a partir de todas las muestras (3.528 turnos, siendo 1.654 turnos de usuario y 1.874 turnos de sistema), los modelos procesados se han estimado a partir de 3.404 muestras (1.591 turnos de usuario y 1.813 turnos de sistema).

Se observa que los nuevos modelos tienen, aproximadamente, la mitad de estados que los modelos iniciales. La reducción en número de las transiciones posibles también es significativa. En el caso del modelo específico, se han mantenido bastantes estados, a pesar de su muy baja frecuencia de aparición en el corpus (64 estados corresponden a situaciones vistas tres o menos veces en el corpus, y, de éstos, 19 a situaciones que aparecen sólo una vez en el corpus). Estos estados no estarán bien estimados, dadas las limitaciones del corpus, pero conviene mantenerlos por cuanto corresponden a actos de diálogo relevantes en la tarea. Por ello, proporcionan alternativas válidas de diálogo, en las situaciones futuras a las que tenga que dar respuesta el gestor de diálogo que use este modelo.

```

Inicializar(RH);           // RH = Registro Histórico
Leer(MD);                 // MD = Modelo de Diálogo
MD.estado = APERTURA;    // estado inicial, Apertura
REPETIR
  Leer(frames-usuario);   // lectura de frames de usuario
  MD.entrada = Adaptar(RH, frames-usuario); // generalización semántica
  MD.estado = Transitar(MD.estado, MD.entrada); // transición a estado de usuario
  RH = Actualizar(RH, frames-usuario); // actualizar RH por turno de
  // usuario
  MD.estado = Transitar(MD.estado, RH); // transición a estado de sistema
  RH = Actualizar(RH, MD.estado, BD.salida); // actualizar RH por turno de
  // sistema
  frames-sistema = Adaptar(MD.estado, RH); // generación de frames de sistema
  Escribir(frames-sistema); // escritura de frames de sistema
HASTA MD.estado = CIERRE // estado final, Cierre

```

Figura 5.4: Algoritmo del gestor de diálogo

5.4 Algoritmo del gestor de diálogo

En la Figura 5.4 se muestra el algoritmo del gestor de diálogo. Cualquier diálogo es una secuencia de turnos entre el usuario y el sistema. El gestor de diálogo realiza un proceso iterativo que consta de los siguientes pasos:

- ❑ Leer los frames o representación semántica del turno de usuario (la salida del módulo de comprensión).
- ❑ Comparar los frames recibidos y los actos de diálogo del usuario, que sean posibles según el modelo de diálogo.
- ❑ Transitar a un estado de usuario en el modelo estocástico de diálogo.
- ❑ Actualizar el registro histórico con los datos proporcionados por el usuario.
- ❑ Transitar a un estado de sistema en el modelo estocástico de diálogo.
- ❑ Actualizar el registro histórico, en el caso de que se consulte la base de datos (cuando se elige un estado de sistema que sea de respuesta).
- ❑ Escribir la representación semántica del turno del sistema (es decir, los frames del sistema que serán las entradas al módulo generador de respuestas).

En esencia, el algoritmo consiste en la gestión de dos componentes: el modelo estocástico de diálogo (MD) y el registro histórico (RH). Mediante estos componentes, el gestor de diálogo decide sus respuestas y, por tanto, establece la estrategia del diálogo.

Como se ha indicado, los modelos estocásticos que usamos son modelos de bigramas. En consecuencia, para cada turno, la decisión de cuál será el siguiente estado de diálogo se toma teniendo en cuenta sólo el turno inmediatamente anterior. Estos modelos adolecen de una insuficiente cobertura del espacio de sucesos de la tarea, debido al reducido tamaño del corpus de entrenamiento. Esto supone un cierto riesgo de no tomar ninguna decisión (ninguna elección de siguiente estado de diálogo) por ausencia de transiciones válidas en el modelo. Se han desarrollado varios procedimientos para prevenir este problema.

En el caso de una transición a un nuevo estado de usuario, se aplica un preproceso a los frames de entrada. Llamamos a este preproceso *generalización semántica*. Además, en el caso de que la generalización semántica no sea ayuda suficiente para encontrar una transición en el modelo específico de diálogo, se usa el modelo genérico de diálogo para buscar alguna transición válida en el modelo específico. La generalización semántica se explicará, ampliamente, en otra sección.

En el caso de una transición a un nuevo estado de sistema, considerar sólo la probabilidad del bigrama no es siempre la mejor decisión. Un gestor de diálogo que ignore toda la información intercambiada con el usuario durante los turnos anteriores podría tomar decisiones inadecuadas. El registro histórico (RH) permite que el gestor de diálogo elija una transición válida en el modelo teniendo en cuenta la información relevante proporcionada en cualquier turno anterior (principalmente, los valores de los atributos que ya sean conocidos). Además, mediante el uso de las medidas de confianza suministradas por el módulo de comprensión, se puede dar más o menos peso a los valores de los atributos almacenados en el RH, y, de este modo, orientar adecuadamente la estrategia del diálogo.

5.5 El registro histórico

El registro histórico es una tabla que contiene tantas filas (registros) como atributos se han definido para la tarea y tantas columnas (campos) como propiedades relevantes se han considerado conveniente usar en el gestor de diálogo. En concreto, el RH almacena, para cada atributo, la siguiente información: momento de modificación (*edad-M*), momento de referencia (*edad-R*), estado de confirmación, confianza asociada y valor del atributo. A continuación, se describe cada uno de estos campos del RH.

El campo que llamamos *edad-M* informa del momento en que se ha modificado el atributo, y lo almacena mediante un número entero relativo a los turnos transcurridos desde tal modificación. En cada actualización del RH, se incrementa en una unidad la *edad-M* de los atributos no modificados y se pone a cero la *edad-M* de los atributos modificados. Dado que se actualiza el RH dos veces por turno de diálogo, la *edad-M* de los atributos no modificados aumenta en 2 unidades por turno. Los atributos que se referencian, pero que no se modifican (es decir, que conservan los valores asignados en la inicialización del RH), tienen una *edad-M* inicial de 80 en lugar de 0. Por tanto, un atributo con $edad-M \geq 80$ es un atributo en uso y que conserva el valor por defecto asignado por el gestor de diálogo.

El campo llamado *edad-R* indica el momento en que ha sido referenciado el atributo. En cada actualización del RH se incrementa en una unidad la *edad-R* de los atributos no referenciados y se pone a cero la *edad-R* de los atributos referenciados. Dado que se actualiza

RH	
edad-M = 81; edad-R = 81; conf = 0; prob = 0.50; valor = Valencia	[ORIGEN]
edad-M = 81; edad-R = 81; conf = 0; prob = 0.50; valor = Zaragoza	[DESTINO]
edad-M = 81; edad-R = 81; conf = 0; prob = 0.00; valor = "hoy"	[FECHA-SALIDA]

Figura 5.5: Contenido inicial del registro histórico

dos veces por turno de diálogo, la *edad-R* de los atributos no referenciados aumenta en 2 unidades por turno.

El campo llamado *conf* guarda el estado de confirmación del atributo. Es un indicador que puede tomar cuatro valores: "0", si el atributo no ha sido confirmado; "1", si el atributo ha sido confirmado; "-1", si el atributo está pendiente de confirmar, mediante confirmación explícita; "-2", si el atributo está pendiente de confirmar, mediante confirmación implícita. Es decir, los dos últimos valores corresponden a una situación temporal, en donde el sistema ha generado un acto de diálogo de confirmación y se está a la espera del siguiente acto de diálogo del usuario. Si el usuario acepta el valor del atributo, su nuevo estado de confirmación será "1". En caso contrario, será "0".

El campo llamado *prob* contiene la confianza asociada al atributo. Es un número real, en el rango [0, 1], que indica la fiabilidad del valor del atributo, almacenado en el RH. Los valores de confianza del RH pueden proceder, directamente, de las confianzas presentes en los frames recibidos, o pueden ajustarse, aplicando determinadas fórmulas, en función de las respuestas afirmativas o negativas del usuario ante las confirmaciones o preguntas del sistema.

El campo llamado *valor* almacena el valor del atributo. Es el dato (número, fecha, cadena de caracteres, etc.) asignado, en el HR, al atributo. Puede proceder de los frames recibidos, si es un dato aportado por el usuario, o de los listados obtenidos de la base de datos, si es el resultado de una consulta generada para obtener la información solicitada por el usuario.

En la Figura 5.5 se muestra el contenido inicial del RH. Para el gestor de diálogo, el viaje por defecto corresponde a un trayecto de Valencia a Zaragoza para la fecha actual.

5.6 Generalización semántica y transiciones a estados de usuario

El corpus de entrenamiento no es bastante grande para estimar el modelo específico de diálogo con una adecuada cobertura del espacio de sucesos de la tarea considerada. En consecuencia, el gestor de diálogo, en su normal funcionamiento, puede encontrarse ante serias limitaciones en la búsqueda de transiciones aceptables en este modelo. A menudo, no es posible establecer una correspondencia directa entre cada frame recibido (frame del usuario) y el acto de diálogo más parecido al frame. En tales casos, se aplica un preproceso a la representación semántica del turno de usuario para así facilitar una entrada más flexible al modelo:

MD.entrada = Adaptar(RH, frames-usuario);
 MD.estado = Transitar(MD.estado, MD.entrada);

La idea que justifica este preproceso, que llamamos *generalización semántica*, es que si se incrementa el número de cadenas (descriptores de actos de diálogo) disponibles para buscar transiciones en el modelo, entonces se incrementa la probabilidad de que la actualización del modelo, la transición a un nuevo estado, tenga lugar. Los frames de usuario se suavizan mediante la función *Adaptar*, que genera un conjunto de cadenas-descriptores de actos de diálogo, al aplicar pequeñas variaciones formales. Así, en lugar de equiparar los frames de usuario con una única cadena-descriptor, se equiparan con *MD-entrada*, un conjunto de descriptores parecidos. Los descriptores del conjunto *MD-entrada* pueden diferir en mayor o menor grado en su forma, pero deben de mantener lo máximo posible del significado de los frames de usuario, especialmente de los frames con los valores de confianza más altos. Se dispone de cinco métodos de generalización o suavizado de la representación semántica del usuario. En dos métodos se tienen en cuenta las medidas de confianza. Los otros tres métodos son independientes de las medidas de confianza.

Por ejemplo, dado un turno de usuario que conste de tres frames:

(AFIRMACIÓN) * [0.78, 0.83]
 DESTINO: Bilbao * [0.82, 0.85]
 TIPO-TREN: Talgo * [0.62, 0.66]
 (HORA-SALIDA) * [0.91, 0.95]
 (PRECIO) * [0.55, 0.46]

Su correspondencia directa, literal, en términos de actos de diálogo es la siguiente:

(U:AFIRMACIÓN:DESTINO,TIPO-TREN)
 (U:PREGUNTA:HORA-SALIDA)
 (U:PREGUNTA:PRECIO)

Algunos ejemplos de las correspondencias que se pueden establecer, con este turno de usuario, al aplicar los diferentes métodos de generalización semántica son los siguientes:

- ❑ Por concatenación incondicional:
 (U:PREGUNTA:HORA-SALIDA) (U:PREGUNTA:PRECIO)
- ❑ Por concatenación condicionada a medidas de confianza:
 (U:AFIRMACIÓN:DESTINO,TIPO-TREN) (U:PREGUNTA:HORA-SALIDA)
- ❑ Por integración:
 (U:PREGUNTA:HORA-SALIDA,PRECIO)
- ❑ Por fragmentación aleatoria:
 (U:AFIRMACIÓN:TIPO-TREN) (U:PREGUNTA:PRECIO)
- ❑ Por fragmentación condicionada a medidas de confianza:
 (U:AFIRMACIÓN:DESTINO) (U:PREGUNTA:HORA-SALIDA)

A continuación, se describen los cinco métodos de generalización semántica:

- ❑ El método de concatenación incondicional. Se construyen nuevas entradas (descriptores de actos de diálogo) para actualizar el modelo de diálogo podando los frames en las posiciones iniciales de la secuencia de entrada. La idea subyacente en este método es que los frames pueden interpretarse como una lista cronológica donde los primeros elementos son menos importantes que los últimos. Así, en el anterior ejemplo, el descriptor (U:PREGUNTA:HORA-SALIDA) (U:PREGUNTA:PRECIO) es la concatenación incondicional de los dos últimos actos de diálogo.

- ❑ El método de concatenación condicionada a medidas de confianza. Como en el método anterior, las nuevas entradas al modelo se generan simplificando la representación semántica original, mediante la poda de frames. Sin embargo, en este método, los frames de entrada se listan en orden ascendente de valores de confianza antes de realizar la poda. Así, en el ejemplo, se concatenan los dos frames con las medidas de confianza más altas, mientras que se ignora el último frame de la secuencia de entrada por ser el que tiene menor confianza.
- ❑ El método de integración. Algunas veces, dos o más frames de usuario pueden corresponder a la misma clase de acto de diálogo (por ejemplo, a una pregunta acerca de ciertos atributos de la tarea). En tales casos, los frames pueden combinarse para buscar su correspondencia con un acto de diálogo cuyo identificador de segundo nivel sea una concatenación de todos los atributos. Este método no ocasiona ninguna pérdida de significado respecto a la entrada original. En el ejemplo anterior, el descriptor (U:PREGUNTA:HORA-SALIDA,PRECIO) coincide con el significado de los dos últimos frames.
- ❑ El método de fragmentación aleatoria. Se basa en la operación inversa a la aplicada en el método de integración. Cuando un frame incluye una lista de atributos más bien larga, disminuye la probabilidad de encontrar una transición en el modelo. En este método, se elige aleatoriamente un subconjunto de los atributos y se construye una nueva entrada con una lista de atributos más corta. Como en los métodos de concatenación, se pierde parte del contenido de la entrada original. En el ejemplo anterior, el descriptor (U:AFIRMACIÓN:TIPO-TREN) (U:PREGUNTA:PRECIO) es el resultado de conservar sólo TIPO-TREN y PRECIO en el subconjunto de atributos.
- ❑ El método de fragmentación condicionada a medidas de confianza. Como en el método de fragmentación aleatoria, las nuevas entradas para actualizar el modelo de diálogo son más cortas porque sólo se mantiene un subconjunto de atributos. La diferencia entre ambos reside en el criterio para generar el subconjunto. En este método, se añaden aquellos atributos cuya confianza sea mayor a un determinado valor umbral. En el último ejemplo, se ha supuesto que 0.8 es el umbral y, por ello, sólo se mantienen los atributos DESTINO y HORA-SALIDA.

Supóngase que el descriptor (U:AFIRMACIÓN:DESTINO) (U:PREGUNTA:HORA-SALIDA), construido mediante fragmentación condicionada a medidas de confianza, determina la transición en el modelo. Entonces, el sistema respondería la consulta sobre horarios en los turnos siguientes. El usuario tendría que repetir la pregunta sobre precios más tarde. Obviamente, esta transición no es la mejor opción.

Para resolver esto, hemos desarrollado una planificación de prioridades para buscar, de un modo razonable, transiciones a estados de usuario. Se ha dado preferencia a las generalizaciones semánticas (descriptores *MD-entrada*) que conserven mejor el significado de los frames de usuario y que tengan los valores de confianza más altos.

Además, los criterios para comparar los descriptores *MD-entrada* y los identificadores de estados de usuario disponibles en MD se relajan progresivamente hasta encontrar una correspondencia. Se han considerado los siguientes tres criterios:

- ❑ Criterio de comparación exacta. La correspondencia entre un descriptor *MD-entrada* y un identificador del estado de usuario se establece sólo si son exactamente iguales.
- ❑ Criterio de inclusión completa. La correspondencia entre un descriptor *MD-entrada* y un identificador del estado de usuario se establece sólo si el descriptor *MD-entrada* está incluido en el identificador del estado de usuario (es decir, si es una subcadena del identificador de estado). Así, por ejemplo, se pueden equiparar un descriptor *MD-entrada* como (U:RESPUESTA:FECHA-SALIDA) y un identificador del estado de usuario como (U:AFIRMACIÓN:DESTINO,ORIGEN) (U:RESPUESTA:FECHA-SALIDA).
- ❑ Criterio mixto de coincidencia e inclusión. La correspondencia entre un descriptor *MD-entrada* y un identificador del estado de usuario se establece si se cumplen dos condiciones: 1ª) coincidencia en primer nivel: los identificadores de primer nivel del etiquetado deben ser los mismos en ambas cadenas; 2ª) inclusión en segundo nivel: todos los identificadores de segundo nivel del etiquetado del estado de usuario deben estar también en el descriptor *MD-entrada* (esta condición es más relajada que en los anteriores criterios porque sólo se tiene en cuenta la presencia de los identificadores, y no su orden de aparición). Así, por ejemplo, se pueden equiparar un descriptor *MD-entrada* como (U:NEGACIÓN:DESTINO,FECHA-SALIDA,ORIGEN) (U:RESPUESTA:DESTINO,FECHA-SALIDA,ORIGEN) y un identificador del estado de usuario (U:NEGACIÓN:DESTINO,ORIGEN) (U:RESPUESTA:DESTINO,ORIGEN).

Las prioridades en la búsqueda de correspondencias entre los identificadores de estados de usuario y los diferentes tipos de generalizaciones semánticas así como la aplicación de los distintos criterios de correspondencia se planifican del siguiente modo:

- ❑ En primer lugar, se aplica el criterio de comparación exacta a los cinco tipos de descriptores de generalización semántica, según el siguiente orden de prioridad: 1º) concatenación condicionada a medidas de confianza, 2º) concatenación incondicional e integración, 3º) fragmentación condicionada a medidas de confianza, y 4º) fragmentación aleatoria. En el momento en que se encuentra una correspondencia, la búsqueda concluye. Así, por ejemplo, sólo se buscarán correspondencias con los descriptores generados por los métodos de concatenación incondicional e integración si no se ha encontrado ninguna con los descriptores generados por concatenación condicionada a medidas de confianza. Y, si se encuentra correspondencia usando los métodos de concatenación o integración, no se considerarán los descriptores generados por el método de fragmentación (al que se le asigna la prioridad más baja al ser el que retiene menos del contenido original de los frames).
- ❑ En segundo lugar, y sólo si no se ha tenido éxito al aplicar el criterio de comparación exacta, se aplica el criterio de inclusión completa. En este caso, todos los descriptores de generalización semántica tienen la misma prioridad. Dada la relajación en la búsqueda al usar este criterio, no se considera relevante primar uno u otro método de generalización semántica. Por tanto, la correspondencia dependerá del orden de exploración de los estados (nodos del modelo).
- ❑ En tercer lugar, y sólo si no se ha tenido éxito al aplicar los dos criterios anteriores, se aplica el criterio mixto de coincidencia e inclusión. También en este caso, los descriptores de generalización semántica tienen la misma prioridad. Por tanto, la

correspondencia dependerá del orden de exploración de los estados (nodos del modelo). Además, sólo se han considerado los descriptores generados por los métodos de concatenación incondicional, integración y fragmentación aleatoria (es decir, los independientes de las medidas de confianza).

En la Figura 5.6 se reproduce un fragmento de diálogo que ilustra el uso de la generalización semántica en la selección del siguiente estado de usuario en MD. En la figura, el estado actual del sistema es (S:PREGUNTA:TIPO-TREN). Por tanto, se genera un frame, y una frase, de pregunta sobre el tipo de tren. Además, se incluyen en el frame atributos tipo ACTUAL, usados para confirmaciones implícitas, que también aparecen en la frase del sistema.

El usuario responde con una negación, referida a algunos de los atributos objeto de confirmación implícita, con una enumeración de los valores correctos (origen, destino, y el tipo de tren preguntado por el sistema) y además repite los objetivos de su consulta (horarios y precios). A partir de los frames recibidos, el gestor de diálogo genera un descriptor de acto de diálogo del usuario, con el que intentará transitar en MD. El descriptor o cadena inicial es la siguiente: (U:NEGACIÓN:TIPO-TREN) (U:PREGUNTA:HORA-SALIDA) (U:PREGUNTA:PRECIO). Desde el estado actual del sistema, hay 6 transiciones disponibles (listadas también en la figura), y ninguno de los correspondientes estados de usuario coincide con la anterior cadena.

En consecuencia, mediante generalización semántica, se procede a construir el conjunto de todas las posibles cadenas similares a la inicial para compararlas con los posibles estados de usuario. En la figura se ha incluido el listado completo, usando las siguientes abreviaturas: *concat.(i.)* = concatenación incondicional; *concat.(m.c.)* = concatenación condicionada a medidas de confianza; *integr.* = integración; *fragm.(a.)* = fragmentación aleatoria; *fragm.(m.c.)* = fragmentación condicionada a medidas de confianza.

La coincidencia se encuentra, después de descartar el primer frame, mediante el método de fragmentación condicionada a medidas de confianza. Este método selecciona el frame de consulta más probable y, por ello, el siguiente estado de usuario es (U:PREGUNTA:HORA-SALIDA).

MD	estado-sistema = (S:PREGUNTA:TIPO-TREN)

(SOLICITUD) 1.00	
TIPO-TREN: alaris 0.54	
ORIGEN-ACTUAL: Lugo 0.71	
DESTINO-ACTUAL: Toledo 0.73	
FECHA-ACTUAL: 24-09-2005 0.61	
S0: Desde Lugo, a Toledo, el día veinticuatro de Septiembre. ¿En qué tipo de tren quiere viajar?	
U1: No, mire, quería ir de Vigo a Oviedo y en alaris, me dicen horarios y precios.	
(NEGACIÓN) 0.73	
(HORA-SALIDA) 0.78	
ORIGEN: Vigo 0.78	
DESTINO: Oviedo 0.74	
TIPO-TREN: alaris 0.53	
(PRECIO) 0.53	
ORIGEN: Vigo 0.78	
DESTINO: Oviedo 0.74	
TIPO-TREN: alaris 0.53	

[U1] = (U:NEGACIÓN:TIPO-TREN)(U:PREGUNTA:HORA-SALIDA)(U:PREGUNTA:PRECIO)	

transiciones disponibles	

prob = 0.0270; (S:PREGUNTA:TIPO-TREN) → (U:PREGUNTA:HORA-SALIDA)	
prob = 0.1351; (S:PREGUNTA:TIPO-TREN) → (U:PREGUNTA:PRECIO)	
prob = 0.7297; (S:PREGUNTA:TIPO-TREN) → (U:RESPUESTA:TIPO-TREN)	
prob = 0.0270; (S:PREGUNTA:TIPO-TREN) → (U:INDEFINIDA:NIL)	
prob = 0.0270; (S:PREGUNTA:TIPO-TREN) → (U:NO_ENTENDIDO:NIL)	
prob = 0.0540; (S:PREGUNTA:TIPO-TREN) → (U:RESPUESTA:TIPO_TREN) (U:PREGUNTA:PRECIO)	

generalizaciones semánticas de [U1]:	

concat.(i)	= [U1]
concat.(mc)	= (U:PREGUNTA:PRECIO)(U:NEGACIÓN:TIPO-TREN)(U:PREGUNTA:HORA-SALIDA)
integr.	= (U:NEGACIÓN:TIPO-TREN)
fragm.(a)	= (U:NEGACIÓN:TIPO-TREN)
fragm.(mc)	= (U:NEGACIÓN:TIPO-TREN)

concat.(i)	= (U:PREGUNTA:HORA-SALIDA)(U:PREGUNTA:PRECIO)
concat.(mc)	= (U:NEGACIÓN:TIPO-TREN)(U:PREGUNTA:HORA-SALIDA)
integr.	= (U:PREGUNTA:HORA-SALIDA,PRECIO)
fragm.(a)	= (U:PREGUNTA:HORA-SALIDA)
fragm.(mc)	= (U:PREGUNTA:HORA-SALIDA)

concat.(i)	= (U:PREGUNTA:PRECIO)
concat.(mc)	= (U:PREGUNTA:HORA-SALIDA)
integr.	= (U:PREGUNTA:PRECIO)
fragm.(a)	= (U:PREGUNTA:PRECIO)
fragm.(mc)	= (U:PREGUNTA:PRECIO)

MD	transición mediante método de fragmentación condicionada a medidas de confianza
MD	estado-usuario = (U:PREGUNTA:HORA-SALIDA)

Figura 5.6. Ejemplo de generalización semántica

5.7 Transiciones a estados de sistema

En el caso de las transiciones a nuevos estados de sistema, el gestor del diálogo ejecuta el siguiente procedimiento. Dados los valores de los atributos y sus medidas de confianza, almacenados en el RH, y el acto de diálogo del usuario, identificado en el estado actual del MD, el gestor de diálogo realiza las siguientes acciones:

- ❑ Construcción de una lista con todas las posibles transiciones desde el actual estado de usuario.
- ❑ Poda de transiciones. Se eliminan de la lista las transiciones que sean incompatibles con el contenido del RH.
- ❑ Reajuste de los pesos de las transiciones permitidas, combinando sus probabilidades, disponibles en el MD, con las medidas de confianza, almacenadas en el RH.
- ❑ Elección de una de las transiciones más probables después del anterior reajuste.

En el segundo paso de este procedimiento, se aplican algunas reglas para podar aquellas transiciones que podrían generar turnos innecesarios o llevar el diálogo hacia una situación problemática (tal que causara confusión en el usuario, y/o el bloqueo del sistema, o su silencio, o una respuesta sin sentido). El conjunto de las reglas de poda de transiciones puede verse como una interpretación del contenido del RH usando el sentido común. Las reglas principales son las siguientes:

- ❑ Podar transiciones a estados de confirmación de atributos si sus valores ya han sido confirmados y/o sus medidas de confianza son mayores a un determinado valor umbral.
- ❑ Podar transiciones a estados de respuestas a las peticiones de los usuarios si alguno de los atributos necesarios para construir la consulta a la base de datos permanece indeterminado (es decir, sin un valor almacenado en el RH) y/o es inseguro (es decir, si su medida de confianza es menor a un determinado valor umbral).

Después, en el tercer paso del procedimiento, las probabilidades de las transiciones permitidas a nuevos estados de sistema se incrementan o disminuyen dependiendo de las medidas de confianza de los atributos implicados en el significado de los posibles nuevos estados de sistema. Las nuevas probabilidades de las transiciones se obtienen mediante:

$$p^*(s_i, s_j) = \mu \cdot p(s_i, s_j) + (1 - \mu) \cdot \min_k (cs_k(s_i, s_j))$$

Donde:

- ❑ $p^*(s_i, s_j)$ es la nueva probabilidad desde el estado s_i al estado s_j .
- ❑ $p(s_i, s_j)$ es la probabilidad inicial de esta transición, dada por el modelo.
- ❑ $cs_k(s_i, s_j)$ es la medida de confianza del atributo k implicado en esta transición.
- ❑ μ es un coeficiente determinado experimentalmente (fijado en 0.7).

De esta manera, por ejemplo, las transiciones a estados de confirmación de atributos con medidas de confianza más bien elevadas (aunque no tanto como para que sean podadas) pueden ser debilitadas, mientras que otras transiciones a confirmar atributos con valores de confianza bajos pueden ser reforzadas. Igualmente, las transiciones a estados de respuesta

pueden debilitarse o reforzarse dependiendo de las medidas de confianza de los atributos que disparan la consulta a la base de datos. Este ajuste de las probabilidades de las transiciones es una modificación temporal, que se mantiene sólo hasta que el cuarto paso del procedimiento se completa. Por tanto, el modelo permanece inalterado una vez que la transición es elegida.

En el cuarto paso de este procedimiento, se genera un umbral aleatorio, relativo a la probabilidad máxima de las transiciones permitidas:

$$umbral_aleatorio = \max (p^*(s_i, s_j)) / factor_aleatorio$$

Donde el denominador se elige, aleatoriamente, en el rango [4, 7]. A continuación, se elige, también de modo aleatorio, una posición en la lista de las transiciones permitidas. El gestor de diálogo escoge la transición que le lleve a un nuevo estado de sistema, recorriendo la lista a partir de esa posición y buscando la primera transición que cumpla que:

$$p^*(s_i, s_j) \geq umbral_aleatorio$$

De esta manera, se priman las transiciones permitidas que tengan probabilidades más altas. La transición con probabilidad máxima será frecuentemente elegida, pero no en exclusiva.

Tal como se acaba de explicar, el RH juega un papel decisivo en la estrategia de selección de transiciones en el MD. Sin embargo, nunca se tiene una certeza absoluta de que el RH no contiene información errónea. Existe, por tanto, algún riesgo de que errores no detectados en el RH causen una elección incorrecta de transición en el MD. Esto no parece crítico en el tercer paso del procedimiento, cuando el RH se usa sólo para aumentar o disminuir las probabilidades de las transiciones permitidas, pero podría ser más problemático en el segundo paso, cuando el RH se consulta para decidir la poda de transiciones.

Suponemos que las medidas de confianza son indicadores fiables de la correcta asignación de valores a los atributos. Entonces, si el valor de un atributo tiene asociado un valor de confianza bajo, se supone que es más bien poco fiable. En consecuencia, la transición que lleve a su confirmación se verá favorecida mientras que cualquier transición a estados de respuesta que necesiten ese atributo será podada. De modo similar, en el caso de tener asociada una medida de confianza alta, se supone que el valor del atributo es correcto, y las transiciones a estados de confirmación del mismo serán podadas.

El funcionamiento satisfactorio del sistema depende de la correcta elección del valor de confianza que servirá de umbral para la poda de transiciones. Este umbral debe determinarse experimentalmente de modo que se minimicen tanto las confirmaciones superfluas como las consultas erróneas a la base de datos (desafortunadamente, se trata de dos objetivos contrapuestos).

5.8 Medidas de confianza y transiciones en el modelo

Como se indicó en la descripción del gestor de diálogo, este módulo recibe los valores de confianza que acompañan a los frames de usuario y los considera como otra fuente de información a la hora de elegir una transición, de entre todas las transiciones disponibles, en el modelo de diálogo. Las medidas de confianza se han usado en dos métodos de generalización semántica y, por tanto, pueden influir en la transición en el modelo por turno de usuario. Además, las medidas de confianza, que se almacenan en el registro histórico junto con los valores de sus respectivos atributos, pueden influir en la transición en el modelo por turno de sistema. Estas dos situaciones se describen a continuación.

Las Figuras 5.7 y 5.8 (véase páginas siguientes) muestran cómo las medidas de confianza pueden cambiar el curso del diálogo e influir en la transición del modelo a estados de usuario. En ambas figuras se parte de la misma situación (un turno de usuario U2). En la Figura 5.7, sin información de confianzas, la transición en el modelo se realiza mediante una concatenación incondicional que mantiene sólo el significado del último frame del usuario. En la Figura 5.8, con información de confianzas, la transición en el modelo se realiza mediante una concatenación condicionada a medidas de confianza que elimina precisamente la información del último frame del usuario debido a su muy bajo valor de confianza.

Consideremos ahora el caso de los turnos de sistema. Cuando hay un conjunto de transiciones posibles a nuevos estados de sistema, las medidas de confianza del turno último o anteriores, que están almacenadas en el registro histórico, pueden ser un factor determinante en la elección de la transición más adecuada, debido a los procedimientos de poda, debilitamiento o refuerzo de transiciones que se describieron anteriormente.

El efecto del uso de las medidas de confianza puede apreciarse en la primera respuesta del sistema (en el turno S0). Sin estas medidas, el sistema casi siempre pide confirmar los atributos origen y destino puesto que la transición a dicho estado es la más probable en el modelo. En la Figura 5.9, se ilustra esta situación.

Sin embargo, teniendo en cuenta las medidas de confianza, la respuesta del sistema puede ser más versátil. Dependiendo de los valores de confianza de los atributos origen, destino y fecha de salida, el gestor de diálogo puede elegir otras transiciones, como muestran las dos alternativas de las Figuras 5.10 y 5.11.

En la Figura 5.10, el sistema responde inmediatamente a la pregunta del usuario porque conoce los valores de los tres atributos, necesarios para realizar la consulta, con valores de confianza suficientemente altos. En la Figura 5.11, el sistema pide una confirmación de la fecha de salida porque hay un valor de confianza bajo asociado a ese atributo mientras que los atributos de las ciudades tienen valores de confianza altos. Finalmente, si se diera el caso contrario (valores de confianza bajos para las ciudades), la transición más probable según el modelo, la confirmación de los atributos origen y destino, vería además reforzada la probabilidad de ser elegida a causa de las medidas de confianza.

U2: *Para ir a Bilbao el dos de Marzo, ¿cuándo sale y llega?, y ¿cuánto es?, por favor.*

U2: (HORA-SALIDA)
 DESTINO: Bilbao
 FECHA-SALIDA: 02-03-2005

(HORA-LLEGADA)
 DESTINO: Bilbao
 FECHA-SALIDA: 02-03-2005

(PRECIO)
 DESTINO: Bilbao
 FECHA-SALIDA: 02-03-2005

...

Transición en MD a estado de usuario: (U:PREGUNTA:PRECIO)
 Transición en MD a estado de sistema: (S:RESPUESTA:PRECIO) (S:NUEVA-CONSULTA:NIL)

...

S2: (INFORMACIÓN)
 PRECIO: 20.00
 (NUEVA-CONSULTA)

S2: *El billete cuesta veinte euros. ¿Desea algo más?*

Figura 5.7: Fragmento de diálogo sin uso de medidas de confianza

U2: *Para ir a Bilbao el dos de Marzo, ¿cuándo sale y llega?, y ¿cuánto es?, por favor.*

U2: (HORA-SALIDA) * [0.90, 0.93]
 DESTINO: Bilbao * [0.90, 0.92]
 FECHA-SALIDA: 02-03-2005 * [0.79, 0.83]

(HORA-LLEGADA) * [0.78, 0.73]
 DESTINO: Bilbao * [0.90, 0.92]
 FECHA-SALIDA: 02-03-2005 * [0.79, 0.83]

(PRECIO) * [0.45, 0.43]
 DESTINO: Bilbao * [0.90, 0.92]
 FECHA-SALIDA: 02-03-2005 * [0.79, 0.83]

...

Transición en MD a estado de usuario: (U:PREGUNTA:HORA-SALIDA,HORA-LLEGADA)
 Transición en MD a estado de sistema: (S:RESPUESTA:HORA-SALIDA,HORA-LLEGADA)
 (S:NUEVA-CONSULTA:NIL)

...

S2: (INFORMACIÓN)
 HORA-SALIDA: 18.00
 HORA-LLEGADA: 20.00
 (NUEVA-CONSULTA)

S2: *El tren sale a las seis de la tarde y llega a las ocho de la tarde. ¿Desea algo más?*

Figura 5.8: Fragmento de diálogo usando medidas de confianza

U0: *Hola, quería viajar de Burgos a Bilbao el dos de Marzo.*
U0: (HORA-SALIDA)
 ORIGEN: Burgos
 DESTINO: Bilbao
 FECHA-SALIDA: 02-03-2005
 ...
 Transición en MD a estado de usuario: (U:PREGUNTA:HORA-SALIDA)
 Transición en MD a estado de sistema: (S:CONFIRMACIÓN:DESTINO,ORIGEN)
 ...
S0: (CONFIRMACIÓN)
 ORIGEN: Burgos
 DESTINO: Bilbao
S0: *¿Quiere viajar desde Burgos a Bilbao?*

Figura 5.9: Fragmento de diálogo sin uso de medidas de confianza

U0: *Hola, quería viajar de Burgos a Bilbao el dos de Marzo.*
U0: (HORA-SALIDA) * [0.97, 0.92]
 ORIGEN: Burgos * [0.97, 0.92]
 DESTINO: Bilbao * [0.97, 0.92]
 FECHA-SALIDA: 02-03-2005 * [0.87, 0.92]
 ...
 Transición en MD a estado de usuario: (U:PREGUNTA:HORA-SALIDA)
 Transición en MD a estado de sistema: (S:RESPUESTA:HORA-SALIDA) (S:NUEVA-CONSULTA:NIL)
 ...
S0: (INFORMACIÓN)
 HORA-SALIDA: 15.30 * [1.00]
 (NUEVA-CONSULTA)
S0: *El tren sale a las tres y media de la tarde. ¿Desea algo más?*

Figura 5.10: Fragmento de diálogo usando medidas de confianza

U0: *Hola, quería viajar de Burgos a Bilbao el dos de Marzo.*
U0: (HORA-SALIDA) * [0.97, 0.92]
 ORIGEN: Burgos * [0.87, 0.92]
 DESTINO: Bilbao * [0.87, 0.92]
 FECHA-SALIDA: 02-03-2005 * [0.36, 0.33]
 ...
 Transición en MD a estado de usuario: (U:PREGUNTA:HORA-SALIDA)
 Transición en MD a estado de sistema: (S:CONFIRMACIÓN:FECHA-SALIDA)
 ...
S0: (CONFIRMACIÓN)
 FECHA-SALIDA: 02-03-2005 * [0.33]
S0: *¿Quiere viajar el dos de Marzo?*

Figura 5.11: Fragmento de diálogo usando medidas de confianza

5.9 Ejemplo de diálogo

A continuación, se muestra y comenta un ejemplo de diálogo. El escenario consiste en obtener horarios de salida y llegada y tipo de tren para un viaje de Valencia a Zaragoza el treinta de Junio por la mañana.

En las figuras, se muestra el contenido del RH que es relevante en el diálogo en curso. Para cada atributo, se indica toda la información que se almacena: *edad-M*, *edad-R*, *conf*, *prob* y *valor*. El contenido del RH se lista después de la recepción de los frames de usuario y después de la selección de nuevo estado de sistema, ya que es en estos dos momentos cuando el RH se actualiza.

En cuanto al MD, se detalla la selección de nuevos estados del sistema, listando las transiciones disponibles, separadas en dos grupos: transiciones podadas y transiciones permitidas. Para cada transición, se indica su probabilidad inicial (determinada por el MD), su probabilidad reajustada (resultado de combinar la probabilidad dada por el MD y las confianzas disponibles en el RH, si es una transición permitida) y el identificador del correspondiente estado de sistema.

En el turno inicial (véase la Figura 5.12, en las páginas siguientes), el usuario pide horarios, facilitando destino y fecha del viaje. Tras recibir el frame, el gestor de diálogo actualiza su RH: el atributo HORA-SALIDA contiene la marca “???” de objetivo de consulta; el atributo FECHA-SALIDA, la fecha suministrada por el usuario; los atributos ORIGEN y DESTINO, los valores por defecto (el origen, sin modificar ni referenciar; el destino, referenciado y no modificado). Todos los atributos están sin confirmar (y, por ello, *conf* = 0, para todos).

En MD, la transición a estado de usuario lleva a (U:PREGUNTA:HORA-SALIDA). La generalización semántica, en casos como éste, donde sólo se recibe un frame, resulta trivial. A continuación, en la transición a estado de sistema, se aprecia que se podan todas las transiciones que llevan a estados de respuesta. Esta poda se produce si alguno de los atributos ORIGEN, DESTINO y FECHA-SALIDA tiene una confianza baja (*prob* < 0.60), situación en la que están los 3 atributos en este momento.

Prácticamente todas las transiciones permitidas contienen un acto de diálogo de confirmación de uno o varios de los atributos ORIGEN, DESTINO y FECHA-SALIDA. Como los 3 atributos tienen confianzas bajas, las probabilidades de estas transiciones se incrementan, como resultado de aplicar la fórmula:

$$p^*(s_i, s_j) = \mu \cdot p(s_i, s_j) + (1 - \mu) \cdot \min_k (cs_k(s_i, s_j))$$

Así, por ejemplo, y teniendo en cuenta que $\mu = 0.7$, para la transición al estado (S:CONFIRMACIÓN:DESTINO,DÍA-SALIDA,ORIGEN):

$$p^* = 0.7 \cdot 0.1259 + 0.3 \cdot (1 - \min(0.50, 0.54, 0.50)) = 0.2381$$

Para elegir una de las transiciones, se ha generado el umbral aleatorio relativo a la probabilidad máxima de las transiciones permitidas (en este caso, umbral = 0.0561 = (1/5) 0.2809). De la transiciones permitidas, sólo la transición a confirmar la consulta de horarios, (S:CONFIRMACION:HORA-SALIDA)(S:ESPERA:NIL), tiene probabilidad inferior a este umbral y se descarta. Entre las restantes, se ha elegido, de modo aleatorio, como nuevo estado

del sistema, (S:CONFIRMACIÓN:DÍA-SALIDA), que corresponde a la tercera transición más probable, en el modelo reajustado temporalmente con $p^*(s_i, s_j)$.

Justo antes de generar el frame de salida, en el RH se marca el atributo FECHA-SALIDA como en fase de confirmación explícita ($conf = -1$) y, tentativamente, se incrementa su confianza ($prob = 0.54 \rightarrow 0.68$) un valor aleatorio en el rango $[0, 0.2]$.

En el segundo turno (véase la Figura 5.13), el usuario acepta la fecha de salida y aporta otra restricción, el intervalo horario. En el RH, además de almacenar el nuevo dato, se marca la FECHA-SALIDA como confirmada ($conf = 1$) y se incrementa su confianza ($prob = 0.82$).

En MD, la transición a estado de usuario lleva a (U:AFIRMACIÓN:DÍA-SALIDA), como resultado de aplicar la generalización semántica, en este caso el método de concatenación condicionada a medidas de confianza (que selecciona el acto de diálogo correspondiente al primer frame por ser el más fiable).

Después, en la transición a estado de sistema, se siguen podando las transiciones a estados de respuesta puesto que de los atributos ORIGEN y DESTINO todavía tienen confianzas bajas ($prob < 0.60$). Por otra parte, tal como establece la fórmula de $p^*(s_i, s_j)$, las transiciones permitidas a confirmaciones de estos atributos aumentan sus probabilidades. Se ha elegido como nuevo estado del sistema, (S:PREGUNTA:DESTINO), a pesar de que su $p(s_i, s_j)$ es igual a cero.

(S: PREGUNTA:DESTINO):
$$p^* = 0.7 \cdot 0.0 + 0.3 \cdot 0.5 = 0.15$$

Pero, aunque la transición tiene probabilidad cero según el MD, el RH le asigna cierta probabilidad, dada la baja confianza del atributo. Luego, como se genera un umbral aleatorio ($=0.0464 = (1/4) 0.1859$) inferior a esa probabilidad, la transición entra en la selección final y resulta elegida.

Como antes en el caso de la fecha, ahora en el RH se marca el atributo DESTINO como en fase de confirmación explícita ($conf = -1$) y, tentativamente, se incrementa su confianza ($prob = 0.50 \rightarrow 0.57$) un valor aleatorio en el rango $[0, 0.2]$. Además, como se selecciona el atributo INTERVALO-HORA-SALIDA para una confirmación implícita, se marca adecuadamente su estado ($conf = -2$).

En el turno siguiente (véase la Figura 5.14), el usuario responde a la pregunta sobre destino y no se refiere al intervalo horario. En consecuencia, en el RH: a) el atributo DESTINO se marca como confirmado ($conf = 1$) y se le asigna la confianza del atributo en el frame ($prob = 0.89$); b) el atributo INTERVALO-HORA-SALIDA, objeto de confirmación implícita y no rechazado por el usuario, incrementa su confianza ($prob = 0.50 \rightarrow 0.66$) un valor aleatorio en el rango $[0.1, 0.4]$, pero no activa su indicador $conf$ (que está reservado para preguntas y confirmaciones explícitas).

En MD, la transición a estado de usuario, mediante trivial generalización semántica, lleva a (U:RESPUESTA:DESTINO). Después, en la transición a estado de sistema, todavía se podan las transiciones a estados de respuesta ya que el atributo ORIGEN mantiene tanto su valor por defecto como su baja confianza inicial. Es interesante observar el reajuste de las probabilidades de las transiciones permitidas:

(S:CONFIRMACIÓN:DESTINO,ORIGEN):

$$p^* = 0.7 \cdot 0.33 + 0.3 \cdot 1 - \min(0.89, 0.50) = 0.3833$$

(S:CONFIRMACIÓN:DESTINO):

$$p^* = 0.7 \cdot 0.33 + 0.3 \cdot (1 - 0.89) = 0.2663$$

Así pues, aunque ambas son válidas con el umbral aleatorio (en este caso, $0.0767 = (1/5) \cdot 0.3833$), la transición que incluye confirmar el dato más inseguro, la ciudad de origen, tendría más probabilidad de ser elegida, como de hecho sucede.

Esta vez, en la actualización del RH previa a la generación de los frames de salida, se marca los atributos ORIGEN y DESTINO como en fase de confirmación explícita ($conf = -1$) y, tentativamente, se incrementa su confianza ($prob = 0.50 \rightarrow 0.61$; $prob = 0.89 \rightarrow 0.93$) un valor aleatorio en el rango $[0, 0.2]$. En cuanto a confirmaciones implícitas, como no queda ningún otro atributo referenciado recientemente ($edad-R \leq 2$) no se incluye ninguna (es decir, ningún atributo tipo ACTUAL en los frames).

En el cuarto turno (véase la Figura 5.15), el usuario responde afirmativamente. Por tanto, en el RH los atributos ORIGEN y DESTINO quedan confirmados ($conf = 1$) y se les asignan una confianza alta (la confianza del frame recibido, $prob = 0.86$).

En MD, la transición a estado de usuario, mediante trivial generalización semántica, lleva a (U:AFIRMACIÓN:DESTINO,ORIGEN). A continuación, en la transición a estado de sistema, se podan las transiciones a estados de pregunta o de confirmación, ya que los valores de los atributos son conocidos y fiables, y también las transiciones a estados de respuesta de atributos no consultados (caso del horario del viaje de vuelta).

Como siempre, para elegir entre las transiciones permitidas, se genera el umbral aleatorio (en este caso, $0.1162 = (1/6) \cdot 0.6973$). Este umbral es superado por dos transiciones: la que contestaría al usuario horarios de salida, exclusivamente; y la que contesta horarios de salida y de llegada. Esta última, pese a su menor probabilidad, resulta elegida. En consecuencia, se agiliza el diálogo, al aportar espontáneamente la hora de llegada, dato que el usuario quería, pero aún no había pedido.

En el quinto turno (véase la Figura 5.16), el usuario pregunta por el tipo de tren, la información que le falta para completar el escenario. En MD, la transición a estado de usuario lleva a (U:PREGUNTA:TIPO-TREN). En la poda de transiciones a estados de sistema, se descartan todas las que corresponden a respuestas sobre atributos que el usuario no ha preguntado (caso de precio) o que ya ha obtenido (caso de los horarios). Finalmente, se elige una transición que supone contestar a la última petición del usuario: (S:ESPERA:NIL) (S:RESPUESTA:TIPO-TREN) (S:NUEVA-CONSULTA:NIL).

Tras esto, en el sexto y último turno de diálogo (que, por su sencillez, no se detalla en figura), el usuario cierra el diálogo. En MD se transita, primero, a (U:CIERRE:NIL), y después, a (S:CIERRE:NIL).

U0: *Quería los horarios a Zaragoza el treinta de Junio.*

(HORA-SALIDA) 0.47

DESTINO: Zaragoza 0.73

FECHA-SALIDA: 30-06-2005 0.54

RH

edad-M = 81; edad-R = 81; conf = 0; prob = 0.50; valor = Valencia

[ORIGEN]

edad-M = 81; edad-R = 1; conf = 0; prob = 0.50; valor = Zaragoza

[DESTINO]

edad-M = 1; edad-R = 1; conf = 0; prob = 0.54; valor = 30-06-2005

[FECHA-SALIDA]

edad-M = 1; edad-R = 1; conf = 0; prob = 0.00; valor = ???

[HORA-SALIDA]

generalización semántica de U0: (U:Pregunta:Hora_salida)

MD estado-usuario = (U:PREGUNTA:HORA-SALIDA)

transiciones podadas:

prob = 0.1871;

(S:Confirmacion:Hora_salida)(S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)

prob = 0.0036; (S:Espera:Nil)(S:Respuesta:Precio)(S:Nueva_consulta:Nil)

prob = 0.1259; (S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)

prob = 0.0072; (S:Nueva_consulta:Nil)

prob = 0.1583; (S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)

prob = 0.0036; (S:Confirmacion:Hora_llegada)

prob = 0.0036; (S:Respuesta:Hora_salida)(S:Pregunta:Tipo_tren)

prob = 0.0072; (S:Confirmacion:Hora_salida)(S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Pregunta:Tipo_tren)

transiciones permitidas:

prob = 0.0683 → 0.1849 ; (S:Confirmacion:Fecha_salida)

prob = 0.1871 → 0.2809 ; (S:Confirmacion:Destino,Origen)

prob = 0.1259 → 0.2381 ; (S:Confirmacion:Destino,Fecha_salida,Origen)

prob = 0.0072 → 0.1550 ; (S:Confirmacion:Origen)

prob = 0.0144 → 0.1601 ; (S:No_entendido:Nil)(S:Confirmacion:Destino,Origen)(S:No_entendido:Nil)

prob = 0.0108 → 0.1576 ; (S:Confirmacion:Destino)

prob = 0.0252 → 0.0176 ; (S:Confirmacion:Hora_salida)(S:Espera:Nil)

prob = 0.0252 → 0.1676 ;

(S:No_entendido:Nil)(S:Confirmacion:Destino,Fecha_salida,Origen)(S:No_entendido:Nil)

prob = 0.0072 → 0.1550 ; (S:No_entendido:Nil)(S:Confirmacion:Destino,Fecha_salida)(S:No_entendido:Nil)

prob = 0.0108 → 0.1576 ; (S:No_entendido:Nil)(S:Confirmacion:Destino,Origen)

prob = 0.0108 → 0.1576 ;

(S:No_entendido:Nil)(S:Confirmacion:Destino,Origen,Tipo_tren)(S:No_entendido:Nil)

prob = 0.0036 → 0.1525 ; (S:Pregunta:Destino)

prob = 0.0036 → 0.1525 ; (S:Confirmacion:Destino)(S:Pregunta:Origen)

máximo = 0.2809; umbral_aleat = 0.0561

MD estado-sistema = (S:CONFIRMACIÓN:FECHA-SALIDA)

RH

edad-M = 82; edad-R = 82; conf = 0; prob = 0.50; valor = Valencia

[ORIGEN]

edad-M = 82; edad-R = 2; conf = 0; prob = 0.50; valor = Zaragoza

[DESTINO]

edad-M = 2; edad-R = 0; conf = -1; prob = 0.68; valor = 30-06-2005

[FECHA-SALIDA]

edad-M = 2; edad-R = 2; conf = 0; prob = 0.00; valor = ???

[HORA-SALIDA]

(CONFIRMACIÓN) 1.00

FECHA-SALIDA: 30-06-2005 0.68

S0: *¿Quiere viajar el treinta de Junio?*

Figura 5.12: Ejemplo de diálogo, 1º turno

U1: *Sí, por la mañana.*
 (AFIRMACIÓN) 0.82
 () 0.50
 INTERVALO-HORA-SALIDA: 05.00-13.00 0.50

RH
 edad-M = 83; edad-R = 83; conf = 0; prob = 0.50; valor = Valencia [ORIGEN]
 edad-M = 83; edad-R = 3; conf = 0; prob = 0.50; valor = Zaragoza [DESTINO]
 edad-M = 3; edad-R = 1; conf = 1; prob = 0.82; valor = 30-06-2005 [FECHA-SALIDA]
 edad-M = 3; edad-R = 3; conf = 0; prob = 0.00; valor = ??? [HORA-SALIDA]
 edad-M = 1; edad-R = 1; conf = 0; prob = 0.50; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]

generalización semántica de U1: (U:AfirMACION:Fecha_salida)
 (U:Respuesta:Intervalo_hora_salida)

MD estado-usuario = (U:AFIRMACION:FECHA-SALIDA)

transiciones podadas:

prob = 0.5385;
 (S:Confirmacion:Hora_salida)(S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)
 prob = 0.0256; (S:Confirmacion:Fecha_salida)
 prob = 0.2564;
 (S:Confirmacion:Hora_salida_v)(S:Espera:Nil)(S:Respuesta:Hora_salida_v)(S:Nueva_consulta:Nil)
 prob = 0.0256; (S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)
 prob = 0.0256; (S:Confirmacion:Hora_salida)(S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Pregunta:Tipo_tren)

transiciones permitidas:

prob = 0.0513 → 0.1859 ; (S:Confirmacion:Destino,Origen)
 prob = 0.0256 → 0.1679 ; (S:Confirmacion:Origen)
 prob = 0.0256 → 0.1679 ; (S:Confirmacion:Destino)
 prob = 0.0256 → 0.0179 ; (S:Confirmacion:Hora_salida)(S:Espera:Nil)

máximo = 0.1859; umbral_aleat = 0.0464
 prob = 0.0000 → 0.1500; (S:Pregunta:Destino)

MD estado-sistema = (S:PREGUNTA:DESTINO)

RH
 edad-M = 84; edad-R = 84; conf = 0; prob = 0.50; valor = Valencia [ORIGEN]
 edad-M = 84; edad-R = 0; conf = -1; prob = 0.57; valor = Zaragoza [DESTINO]
 edad-M = 4; edad-R = 2; conf = 1; prob = 0.82; valor = 30-06-2005 [FECHA-SALIDA]
 edad-M = 4; edad-R = 4; conf = 0; prob = 0.00; valor = ??? [HORA-SALIDA]
 edad-M = 2; edad-R = 2; conf = -2; prob = 0.50; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]

(SOLICITUD) 1.00
 DESTINO: Zaragoza 0.57
 INTERVALO-HORA-SALIDA-ACTUAL: 05.00-13.00 0.50
S1: *Salida por la mañana, ¿a dónde quiere ir?*

Figura 5.13: Ejemplo de diálogo, 2º turno

U2: A Zaragoza.

() 0.89

DESTINO: Zaragoza 0.89

RH

edad-M = 85; edad-R = 85; conf = 0; prob = 0.50; valor = Valencia

[ORIGEN]

edad-M = 85; edad-R = 1; conf = 1; prob = 0.89; valor = Zaragoza

[DESTINO]

edad-M = 5; edad-R = 3; conf = 1; prob = 0.82; valor = 30-06-2005

[FECHA-SALIDA]

edad-M = 5; edad-R = 5; conf = 0; prob = 0.00; valor = ???

[HORA-SALIDA]

edad-M = 3; edad-R = 3; conf = 0; prob = 0.66; valor = 05.00-13.00

[INTERVALO-
HORA-SALIDA]

generalización semántica de U2: (U:Respuesta:Destino)

MD estado-usuario = (U:RESPUESTA:DESTINO)

transiciones podadas:

prob = 0.1667 → 0.1167 ;

(S:Confirmacion:Hora_llegada)(S:Espera:Nil)(S:Respuesta:Hora_llegada)(S:Nueva_consulta:Nil)

prob = 0.1667 → 0.1167 ;

(S:Confirmacion:Hora_salida)(S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)

transiciones permitidas:

prob = 0.3333 → 0.3833 ; (S:Confirmacion:Destino,Origen)

prob = 0.3333 → 0.2663 ; (S:Confirmacion:Destino)

máximo = 0.3833; umbral_aleat = 0.0767

MD estado-sistema = (S:CONFIRMACIÓN:DESTINO,ORIGEN)

RH

edad-M = 86; edad-R = 0; conf = -1; prob = 0.61; valor = Valencia

[ORIGEN]

edad-M = 86; edad-R = 2; conf = -1; prob = 0.93; valor = Zaragoza

[DESTINO]

edad-M = 6; edad-R = 4; conf = 1; prob = 0.82; valor = 30-06-2005

[FECHA-SALIDA]

edad-M = 6; edad-R = 6; conf = 0; prob = 0.00; valor = ???

[HORA-SALIDA]

edad-M = 4; edad-R = 4; conf = 0; prob = 0.66; valor = 05.00-13.00

[INTERVALO-
HORA-SALIDA]

(CONFIRMACIÓN) 1.00

ORIGEN: Valencia 0.61

DESTINO: Zaragoza 0.93

S2: ¿Quiere ir de Valencia a Zaragoza?

Figura 5.14: Ejemplo de diálogo, 3º turno

U3: Sí.

(AFIRMACIÓN) 0.86

RH

edad-M = 87; edad-R = 1; conf = 1; prob = 0.86; valor = Valencia	[ORIGEN]
edad-M = 87; edad-R = 3; conf = 1; prob = 0.86; valor = Zaragoza	[DESTINO]
edad-M = 7; edad-R = 5; conf = 1; prob = 0.82; valor = 30-06-2005	[FECHA-SALIDA]
edad-M = 7; edad-R = 7; conf = 0; prob = 0.00; valor = ???	[HORA-SALIDA]
edad-M = 5; edad-R = 5; conf = 0; prob = 0.66; valor = 05.00-13.00	[INTERVALO-HORA-SALIDA]

generalización semántica de U3: (U:AfirMACION:Destino,Origen)

MD estado-usuario = (U:AFIRMACIÓN:DESTINO,ORIGEN)

transiciones podadas:

prob = 0.0270; (S:Confirmacion:Hora_salida_v)(S:Espera:Nil)
 prob = 0.0135; (S:Pregunta:Fecha_salida)
 prob = 0.0135; (S:Confirmacion:Destino,Origen)
 prob = 0.0405;
 (S:Confirmacion:Hora_salida_v)(S:Espera:Nil)(S:Respuesta:Hora_salida_v)(S:Nueva_consulta:Nil)
 prob = 0.2973; (S:Confirmacion:Fecha_salida)

transiciones permitidas:

prob = 0.5676 → 0.6973;
 (S:Confirmacion:Hora_salida)(M:Espera:Nil)(S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)
 prob = 0.0135 → 0.3094;
 (S:Confirmacion:Hora_llegada,Hora_salida)(S:Espera:Nil)(S:Respuesta:Hora_llegada,Hora_salida)(S:Nueva_consulta:Nil)
 prob = 0.0270 → 0.0189; (S:Confirmacion:Hora_salida)(S:Espera:Nil)

máximo = 0.6973; umbral_aleat = 0.1162

MD estado-sistema =

(S:CONFIRMACION:HORA-LLEGADA,HORA-SALIDA)(S:ESPERA:NIL)(S:RESPUESTA:HORA-LLEGADA,HORA-SALIDA)(S:NUEVA-CONSULTA:NIL)

RH

edad-M = 88; edad-R = 2; conf = 1; prob = 0.86; valor = Valencia	[ORIGEN]
edad-M = 88; edad-R = 4; conf = 1; prob = 0.86; valor = Zaragoza	[DESTINO]
edad-M = 0; edad-R = 0; conf = 0; prob = 0.99; valor = 02	[NÚMERO-TRENES]
edad-M = 8; edad-R = 6; conf = 1; prob = 0.82; valor = 30-06-2005	[FECHA-SALIDA]
edad-M = 0; edad-R = 0; conf = 0; prob = 0.99; valor = 06.35	[HORA-SALIDA]
edad-M = 0; edad-R = 0; conf = 0; prob = 0.99; valor = 09.35	[HORA-LLEGADA]
edad-M = 6; edad-R = 6; conf = 0; prob = 0.66; valor = 05.00-13.00	[INTERVALO-HORA-SALIDA]

(CONFIRMACIÓN) 1.00

HORA-SALIDA: ??? 0.00

HORA-LLEGADA: ??? 0.00

(ESPERA) 1.00

(INFORMACIÓN) 1.00

NÚMERO-TRENES: 02 0.99

HORA-SALIDA: 06.35 0.99

HORA-LLEGADA: 09.35 0.99

(NUEVA-CONSULTA) 1.00

S3: Hay 2 trenes, y le informo del primero. Le consulto horarios de salida y de llegada. Un momento por favor. El tren sale a las siete menos veinticinco de la mañana y llega a las diez menos veinticinco de la mañana. ¿Desea algo más?

Figura 5.15: Ejemplo de diálogo, 4º turno

U4: *¿Qué tipo de tren es?*

(TIPO-TREN) 0.50

RH

edad-M = 89; edad-R = 3; conf = 1; prob = 0.86; valor = Valencia
 edad-M = 89; edad-R = 5; conf = 1; prob = 0.86; valor = Zaragoza [
 edad-M = 1; edad-R = 1; conf = 0; prob = 0.00; valor = ???
 edad-M = 1; edad-R = 1; conf = 0; prob = 0.99; valor = 02
 edad-M = 9; edad-R = 7; conf = 1; prob = 0.82; valor = 30-06-2005
 edad-M = 1; edad-R = 1; conf = 0; prob = 0.99; valor = 06.35
 edad-M = 1; edad-R = 1; conf = 0; prob = 0.99; valor = 09.35
 edad-M = 7; edad-R = 7; conf = 0; prob = 0.66; valor = 05.00-13.00

[ORIGEN]
 [DESTINO]
 [TIPO-TREN]
 [NÚMERO-TRENES]
 [FECHA-SALIDA]
 [HORA-SALIDA]
 [HORA-LLEGADA]
 [INTERVALO-
 HORA-SALIDA]

generalización semántica de U4: (U:Pregunta:Tipo_tren)

MD estado-usuario = (U:PREGUNTA:TIPO-TREN)

transiciones podadas:

prob = 0.0200; (S:Espera:Nil)(S:Respuesta:Precio)(S:Nueva_consulta:Nil)
 prob = 0.0200; (S:Respuesta:Precio)(S:Nueva_consulta:Nil)
 prob = 0.1600; (S:Pregunta:Tipo_tren)
 prob = 0.0200; (S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)
 prob = 0.0200; (S:Espera:Nil)(S:Respuesta:Hora_llegada)(S:Nueva_consulta:Nil)
 prob = 0.0400; (S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)
 prob = 0.0200; (S:Confirmacion:Hora_salida)(S:Espera:Nil)
 prob = 0.1200; (S:Espera:Nil)(S:Pregunta:Tipo_tren)
 prob = 0.0400; (S:Respuesta:Hora_salida)(S:Pregunta:Tipo_tren)
 prob = 0.0600; (S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Pregunta:Tipo_tren)

transiciones permitidas:

prob = 0.3000 → 0.5100 ; (S:Espera:Nil)(S:Respuesta:Tipo_tren)(S:Nueva_consulta:Nil)
 prob = 0.1400 → 0.3980 ; (S:Respuesta:Tipo_tren)(S:Nueva_consulta:Nil)
 prob = 0.0400 → 0.0280 ; (S:Confirmacion:Tipo_tren)

máximo = 0.5100; umbral_aleat = 0.0729

MD estado-sistema = (S:ESPERA:NIL)(S:RESPUESTA:TIPO-TREN)(S:NUEVA-CONSULTA:NIL)

RH

edad-M = 90; edad-R = 4; conf = 1; prob = 0.86; valor = Valencia
 edad-M = 90; edad-R = 6; conf = 1; prob = 0.86; valor = Zaragoza
 edad-M = 0; edad-R = 0; conf = 0; prob = 0.99; valor = talgo
 edad-M = 2; edad-R = 2; conf = 0; prob = 0.99; valor = 02
 edad-M = 10; edad-R = 8; conf = 1; prob = 0.82; valor = 30-06-2005
 edad-M = 2; edad-R = 2; conf = 0; prob = 0.99; valor = 06.35
 edad-M = 2; edad-R = 2; conf = 0; prob = 0.99; valor = 09.35
 edad-M = 8; edad-R = 8; conf = 0; prob = 0.66; valor = 05.00-13.00

[ORIGEN]
 [DESTINO]
 [TIPO-TREN]
 [NÚMERO-TRENES]
 [FECHA-SALIDA]
 [HORA-SALIDA]
 [HORA-LLEGADA]
 [INTERVALO-
 HORA-SALIDA]

(ESPERA) 1.00

(INFORMACIÓN) 1.00

TIPO-TREN: talgo 0.99

(NUEVA-CONSULTA) 1.00

S4: *Un momento por favor. Es un tren talgo. ¿Desea algo más?*

Figura 5.16: Ejemplo de diálogo, 5º turno

6. Generador de respuestas

6. Generador de respuestas

El módulo generador de respuestas traduce las representaciones semánticas de los turnos del sistema (la salida del módulo gestor de diálogo) a frases en español usando plantillas y reglas de combinación de estas plantillas con los valores de los atributos. Cuando el sistema responde a una consulta, los atributos proporcionados por la base de datos han de incluirse en la respuesta del sistema. Además, otros atributos podrían incluirse también, especialmente los que hayan cambiado su valor en el último turno del usuario.

En esta sección se presenta el módulo generador de respuestas del sistema diseñado. En primer lugar, se describe el formato de la información de entrada al módulo, es decir, los frames del sistema, generados por el gestor de diálogo. A continuación, se describe el algoritmo del generador de respuestas y se ilustra su funcionamiento con algunos ejemplos.

6.1 Descripción de los frames del sistema

Los frames de respuesta del sistema se construyen, en el módulo gestor de diálogo, a partir del estado actual del modelo de diálogo, completando la información de valores de atributos mediante la consulta del registro histórico. En la Tabla 6.1 se lista los identificadores de frame del sistema y su correspondencia con los actos de diálogo (con el 1º nivel del modelo de diálogo):

<i>Identificador de frame</i>	<i>1º nivel etiquetado acto de diálogo</i>
(APERTURA)	Apertura
(CIERRE)	Cierre
(CONFIRMACIÓN)	Confirmación
(ESPERA)	Espera
(INFORMACIÓN)	Respuesta
(NO-ENTENDIDO)	No-Entendido
(NUEVA-CONSULTA)	Nueva-Consulta
(SOLICITUD)	Pregunta

Tabla 6.1. Frames del sistema

Todos los frames del sistema van seguidos de su valor de confianza, que es siempre 1.00, porque este valor, actualmente, no se usa. Los frames (CONFIRMACIÓN), (INFORMACIÓN) y (SOLICITUD) irán seguidos por los pares nombre y valor de atributo que corresponda según el estado alcanzado en el modelo de diálogo.

Además, y para permitir que el generador de respuestas pueda realizar confirmaciones implícitas, los frames del sistema pueden ir seguidos, también, por los pares nombre y valor de atributo de tipo ACTUAL que estén presentes en el registro histórico del gestor de diálogo y hayan sido modificados en el último turno.

Los nombres de atributo son los siguientes: TIPO-VIAJE, ORIGEN, DESTINO, TIPO-TREN, CLASE-BILLETE, NÚMERO-RELATIVO-ORDEN, PRECIO, TRAYECTO, TIEMPO-RECORRIDO, SERVICIO, NÚMERO-TRENES, FECHA-SALIDA, FECHA-LLEGADA, INTERVALO-FECHA-SALIDA, INTERVALO-FECHA-LLEGADA, HORA-SALIDA, HORA-LLEGADA, INTERVALO-HORA-SALIDA, INTERVALO-HORA-LLEGADA.

Aunque, dado el actual modelo de diálogo, los atributos usados en los frames del sistema son sólo los siguientes: ORIGEN, DESTINO, TIPO-TREN, PRECIO, TIEMPO-RECORRIDO, SERVICIO, NÚMERO-TRENES, FECHA-SALIDA, HORA-SALIDA, HORA-LLEGADA.

Los nombres de atributo para el viaje de vuelta son los mismos seguidos del sufijo -V. Los nombres de atributo de tipo ACTUAL son los mismos seguidos del sufijo -ACTUAL.

A continuación, se ilustra el formato de los frames más importantes, los frames (CONFIRMACIÓN), (INFORMACIÓN) y (SOLICITUD), mediante algunos ejemplos:

- ❑ Supongamos que (S:CONFIRMACIÓN:DESTINO,DÍA_SALIDA,ORIGEN) es el estado alcanzado en el modelo de diálogo. Entonces, el frame generado por el sistema es:

```
(CONFIRMACIÓN) 1.00
    ORIGEN: ccc d.dd
    DESTINO: ccc d.dd
    FECHA-SALIDA: dd-dd-dddd d.dd
```

Es decir, el nombre del atributo va seguido del carácter “:”, del valor del atributo, de un espacio en blanco, y del valor de la confianza. En el ejemplo, *ccc* indica secuencia de caracteres (en este caso, los nombres de las ciudades en el registro histórico del gestor de diálogo), *dd-dd-dddd* indica formato de fecha (el valor se tomaría del registro), y *d.dd* indica un número real con dos decimales que representa el valor de confianza del atributo (igualmente, el valor se tomaría del registro).

- ❑ Supongamos ahora que (S:CONFIRMACIÓN:HORA-LLEGADA) es el estado alcanzado. Entonces el frame del sistema puede ser:

```
(CONFIRMACIÓN) 1.00
    { HORA-LLEGADA: dd.dd 1.00 }
```

Donde lo encerrado entre llaves puede no aparecer (aparece sólo si el gestor de diálogo dispone de una hora de llegada en su registro).

Pero, además, el frame del sistema puede ser:

```
(CONFIRMACIÓN) 1.00
    HORA-LLEGADA: ??? 0.00
```

Cuando el valor del atributo almacenado en el registro sea, justamente, “???” literal que indica que se trata de un atributo preguntado por el usuario y cuyo valor aún no ha sido obtenido (porque aún no se ha consultado la base de datos).

- ❑ Si el estado alcanzado es (S:PREGUNTA:TIPO-TREN), entonces el frame del sistema puede ser:

```
(SOLICITUD) 1.00
```

TIPO-TREN: *ccc d.dd*
 ORIGEN-ACTUAL: *ccc d.dd*
 INTERVALO-HORA-SALIDA-ACTUAL: *dd.dd-dd.dd d.dd*

Suponiendo que los valores actuales de ciudad de origen e intervalo de hora de salida se hubieran modificado en el último turno (y pudiera ser conveniente citarlos en la respuesta, como confirmación implícita, en una frase como, por ejemplo: “*desde Valencia, por la tarde, ¿en qué tipo de tren quiere viajar?*”).

- Si el estado alcanzado es (S:RESPUESTA:HORA-LLEGADA,HORA-SALIDA), entonces el frame del sistema puede ser:

(INFORMACIÓN) 1.00
 HORA-SALIDA: *dd.dd 1.00*
 HORA-LLEGADA: *dd.dd 1.00*
 FECHA-ACTUAL: *dd-dd-dddd d.dd*

Suponiendo que el valor actual de la fecha de salida se hubiera modificado en el último turno (y pudiera ser conveniente citarlo en la respuesta, como confirmación implícita, mediante una frase como la siguiente: “*el día tres de junio, el tren sale a las cinco de la tarde y llega a las nueve de la noche*”). El valor de confianza de los atributos extraídos de la base de datos, es siempre 1.00, porque este valor, actualmente, no se usa.

Como referencia, la Tabla 10.1 del Anexo lista todos los estados del sistema en el actual modelo de diálogo, y su correspondiente secuencia de frames.

6.2 Descripción del generador de respuestas

En la Figura 6.1 se muestra el algoritmo del generador de respuestas. Se basa en el uso de un diccionario de plantillas de frases, de donde se extraen las plantillas que corresponda según los frames recibidos y, para construir una frase, se rellenan con los valores de los atributos encontrados en los frames.

Con más detalle, se puede describir la iteración que realiza el generador de respuestas en cada turno de diálogo del modo siguiente:

- Lectura de los frames del sistema (la salida del gestor de diálogo).
- Seleccionar las plantillas de frase adecuadas. Para cada frame recibido, se busca, en el diccionario de plantillas de frases, la plantilla que corresponda al frame. Las plantillas seleccionadas se guardan en una lista de segmentos para construir, más tarde, la frase de respuesta del sistema.
- Extraer los valores de los atributos. Todos los valores de atributos presentes en los frames (excepto los de tipo –ACTUAL) se guardan en la lista de los atributos llamados “normales”. Después se usarán para rellenar las plantillas seleccionadas en el punto anterior.

```

Leer(diccionarios-atributos);      // leer diccionarios de valores de atributos
Leer(diccionario-plantillas);      // leer diccionario de plantillas de frases a generar
REPETIR
  Inicializar(listas-atributos);    // listas de atributos vacías
  Inicializar(frase-sistema);       // frase del sistema = cadena vacía
  Leer(frames-sistema);              // lectura de frames de sistema

  // identificar plantillas correspondientes a los frames recibidos
  plantilla-frase = Comparar(diccionario-plantillas, frames-sistema);

  // extraer todos los valores de los atributos presentes en los frames
  atributos-normales = ExtraerTodos(frames-sistema);

  // extraer los valores de los atributos tipo ACTUAL con confianza < umbral
  atributos-actuales = ExtraerInseguros(frames-sistema, umbral);

  // construir el segmento de frase correspondiente a atributos tipo ACTUAL
  segmento-atributos-actuales = Rellenar(plantillas-actuales, atributos-actuales);

  // construir el segmento principal de la frase del sistema
  segmento-principal = Rellenar(plantilla-frase, atributos-normales);

  // concatenar los segmentos para generar la frase del sistema
  frase-sistema = Concatenar(segmento-atributos-actuales, segmento-principal);
  Escribir(frase-sistema);          // escritura de frases de sistema
HASTA frame-sistema = CIERRE      // estado final, Cierre

```

Figura 6.1: Algoritmo del generador de respuestas

- ❑ Extraer atributos “actuales” para confirmación implícita. Los valores de los atributos de tipo –ACTUAL que aparezcan en los frames con una confianza menor a un umbral (fijado en 0.80), se guardan en la lista de los atributos “actuales”.
- ❑ Construir el segmento de confirmación implícita. Las plantillas correspondientes a los atributos en la lista de los atributos “actuales” se seleccionan, rellenan y concatenan para formar este segmento. De este modo, los valores de los atributos, almacenados en el registro histórico del gestor de diálogo, que se hayan modificados en el último turno y que tengan baja confianza, serán citados en la frase de respuesta del sistema a modo de confirmación implícita.
- ❑ Construir el segmento principal de la frase. Es el resultado de rellenas las plantillas de frase seleccionadas con los valores guardados en la lista de atributos “normales”.
- ❑ Construir la frase del sistema. Es decir, concatenar el segmento de confirmación implícita y el segmento principal de la frase.

Al final de cada iteración, la frase del sistema se escribe para su uso por el módulo de síntesis de voz (en el caso de que en el sistema de diálogo se haya integrado este módulo) o por un módulo interfaz que muestre en pantalla la respuesta del sistema (en el caso de que se utilice el sistema de diálogo con entrada y salida de tipo texto).

Como referencia, la Tabla 10.2 del Anexo lista todas las plantillas de frases y sus correspondencias con los frames de sistema.

Ilustraremos con un par de ejemplos el procedimiento de generación de frases que se acaba de exponer.

En primer lugar, supongamos que, tras alcanzar en su modelo de diálogo el estado (S:CONFIRMACIÓN:DESTINO) (S:PREGUNTA:ORIGEN), el gestor de diálogo ha generado los siguientes frames:

```
(CONFIRMACIÓN) 1.00
  DESTINO: Alicante 0.85
  FECHA-SALIDA-ACTUAL: 29-05-???? 0.63
  INTERVALO-HORA-SALIDA-ACTUAL: 13.00-21.00 0.84
(SOLICITUD) 1.00
  ORIGEN: Valencia 0.50
  FECHA-SALIDA-ACTUAL: 29-05-???? 0.63
  INTERVALO-HORA-SALIDA-ACTUAL: 13.00-21.00 0.84
```

Donde, además de incluir los atributos obligatorios ORIGEN y DESTINO, también se han incluido los atributos FECHA-SALIDA e INTERVALO-HORA-SALIDA, de tipo ACTUAL, al considerar en este supuesto que los valores de estos atributos se modificaron en la intervención previa del usuario.

Dado que el generador de respuestas recibe dos frames, ejecutará dos veces la búsqueda en su diccionario de plantillas de frases. Para hallar una plantilla que se adecue al primer frame, usará como criterios de búsqueda los identificadores (CONFIRMACIÓN) y DESTINO. Encontrará la plantilla “¿ quiere ir a <Destino> ?”. En el caso del segundo frame, buscará usando los criterios (SOLICITUD) y ORIGEN, encontrando la plantilla “¿ de dónde quiere salir ?”.

Después, en la lista de atributos normales, se guardarán los valores *Alicante* y *Valencia* como destino y origen, respectivamente. Además, como el valor de la fecha de salida, 29-05-????, aparece con un valor de confianza, 0.63, inferior al umbral de 0.80, se guardará también, en la lista de atributos actuales. En cambio, el valor del intervalo horario, que figura con confianza superior al umbral, se ignorará.

El llamado segmento de atributos actuales contendrá, simplemente, la transcripción en español de la fecha de salida: “el veintinueve de mayo ,”. El segmento principal de la frase será el resultado de concatenar la plantilla “¿ quiere ir a <Destino> ?” y la plantilla “¿ de dónde quiere salir ?” y, a continuación, sustituir las variables que aparecen en el segmento por los correspondientes valores almacenados en la lista de atributos. En este caso, sólo se incorpora a la frase el valor de la ciudad destino, y no el de la ciudad origen dado que esta última no tiene ninguna posición reservada en el segmento.

Por tanto, la frase que el generador de respuestas construirá a partir de los frames recibidos es la siguiente: “el veintinueve de mayo , ¿ quiere ir a Alicante ? ¿ de dónde quiere salir ?”. Frase donde se combina la confirmación explícita de destino y la pregunta de origen, correspondientes al estado alcanzado en el modelo de diálogo, con la confirmación implícita de la fecha de salida, debido al uso del registro histórico y las medidas de confianza.

Como segundo ejemplo, supongamos que, tras alcanzar en su modelo de diálogo el estado (S:CONFIRMACIÓN:HORA-LLEGADA,HORA-SALIDA) (S:ESPERA:NIL) (S:RESPUESTA:HORA-LLEGADA,HORA-SALIDA) (S:NUEVA-CONSULTA:NIL), el gestor de diálogo ha generado los siguientes frames:

```
(CONFIRMACIÓN) 1.00
    HORA-SALIDA: ??? 1.00
    HORA-LLEGADA: ??? 1.00
    ORIGEN-ACTUAL: Zaragoza 0.75
    TIPO-TREN-ACTUAL: alaris 0.70
(ESPERA) 1.00
(INFORMACIÓN) 1.00
    HORA-SALIDA: 11.45 1.00
    HORA-LLEGADA: 14.20 1.00
    ORIGEN-ACTUAL: Zaragoza 0.75
    TIPO-TREN-ACTUAL: alaris 0.70
(NUEVA-CONSULTA) 1.00
```

Donde, además de incluir los atributos obligatorios HORA-SALIDA y HORA-LLEGADA, también se han incluido los atributos ORIGEN y TIPO-TREN, de tipo ACTUAL, al considerar en este supuesto que los valores de estos atributos se modificaron en la intervención previa del usuario.

Como el generador de respuestas recibe cuatro frames, ejecutará cuatro veces la búsqueda en su diccionario de plantillas de frases. La plantilla que corresponde al frame (ESPERA) es “*un momento por favor .*” y la plantilla que corresponde al frame (NUEVA-CONSULTA) es “*¿ desea algo más ?*”.

Para hallar una plantilla adecuada al primer frame, el generador de respuestas usa como criterios de búsqueda los identificadores (CONFIRMACIÓN), HORA-SALIDA y HORA-LLEGADA. Encuentra la plantilla “*le consulto horarios de salida y de llegada .*”. Hay que destacar que existen otras plantillas parecidas que cumplen dos de los tres criterios de búsqueda, pero sólo una, la elegida, satisface la condición completa.

En el caso del tercer frame, buscará usando los criterios (INFORMACIÓN), HORA-SALIDA y HORA-LLEGADA, encontrando la plantilla “*el tren sale a <Hora-Salida> y llega a <Hora-Llegada> .*”.

A continuación, en la lista de atributos normales, se guardarán los valores *11.45* y *14.20* como horas de salida y de llegada, respectivamente. Además, en la lista de atributos actuales, se guardarán los valores de la ciudad origen, *Zaragoza*, y del tipo de tren, *alaris*, puesto que ambos aparecen con confianzas inferiores al umbral.

El segmento de atributos actuales se construirá mediante la concatenación de dos plantillas: “*desde <Origen> ,*” y “*en <Tipo-Tren> ,*” que serán instanciadas con los valores *Zaragoza* y *alaris*, respectivamente. De manera análoga, el segmento principal de la frase se genera concatenando las cuatro plantillas ya citadas e instanciando las variables con las oportunas expresiones en español de las horas.

Así pues, la frase que el generador de respuestas construirá en este segundo ejemplo es la siguiente: “*desde Zaragoza , en alaris , le consulto horarios de salida y de llegada . un*

momento por favor . el tren sale a las doce menos cuarto de la mañana y llega a las dos y veinte de la tarde . ¿ desea algo más ? ”.

6.3 Confirmación implícita y medidas de confianza

La confirmación implícita constituye una estrategia interesante para que el sistema asegure la validez de los datos que el usuario ha proporcionado en los turnos precedentes. Representa, además, una alternativa a la estrategia de confirmación explícita, pues permite verificar los atributos a un menor coste en términos de duración del diálogo. La confirmación implícita se puede combinar con cualquiera de las respuestas del sistema, tal como se ha visto en los dos ejemplos anteriores.

Las cuestiones más importantes que se plantean al realizar confirmaciones implícitas son *¿cuándo confirmar?* y *¿qué confirmar?* Una política del sistema muy conservadora sería incluir en todos los turnos de sistema confirmaciones implícitas de todos los atributos modificados en el anterior turno de usuario (para ser más precisos, que el sistema creyera que se hubieran modificado). Si el usuario acostumbrara a citar bastantes atributos, esta política sería bastante tediosa y no está claro que fuese la más segura. El tedio puede facilitar la falta de atención del usuario, y un atributo incorrecto, objeto de confirmación implícita y no rechazado por el usuario, se transforma en válido para el sistema.

Al igual que el módulo de comprensión, el gestor de diálogo proporciona un conjunto de frames donde, para cada atributo incorporado al frame, tanto su valor como su confianza son leídos del registro histórico y facilitados al módulo generador de respuestas. Esto, como se ha visto, nos permite generar respuestas en lenguaje natural que sean bastante detalladas, y que incluyan o excluyan ciertos atributos, dependiendo de sus confianzas. Éste es el criterio con que se aborda la cuestión *¿qué confirmar?* y, por tanto, se determina la estrategia de confirmación implícita.

Según las medidas de confianza de los atributos opcionales (los de tipo ACTUAL) sean altas o bajas, el generador de respuestas puede decidir omitirlos o mencionarlos. Si muchos atributos opcionales tienen valores de confianza bajos (inferiores a un cierto umbral), estarán presentes en la respuesta. En cambio, cuando los valores de estos atributos sean bastante fiables, la referencia a los mismos se podrá y la respuesta del sistema será tan breve como determine el acto de diálogo a que corresponda: pregunta o confirmación explícita de atributos, o resultado de la consulta a la base de datos.

Hemos supuesto que las medidas de confianza son indicadores fiables de la validez de los valores de los atributos. Desde esta perspectiva, el procedimiento de confirmación implícita incorporado en el módulo de generación de respuestas puede facilitar la corrección de errores. Así, el usuario podrá detectar errores de comprensión

por parte del sistema y, en consecuencia, tendrá la oportunidad de corregirlos, encauzando el diálogo hacia la correcta resolución de su consulta.

En la Figura 6.2 se muestra un fragmento de diálogo (tres turnos de usuario y tres turnos de sistema) donde se ilustra la posibilidad de corrección de errores gracias a la confirmación implícita. Como sabemos, el generador de respuestas filtra los atributos ACTUAL que tengan una confianza superior al umbral 0.80. En el ejemplo, en el turno U3, el usuario ha modificado los valores de la fecha de salida (*cuatro de Marzo*) y del intervalo horario (*por la tarde*), y estos atributos se han añadido a la representación semántica del sistema como atributos ACTUAL. Sin embargo, se ha producido un error, en reconocimiento del habla o en comprensión, y el valor de la fecha de salida (*cuatro de Mayo*) que supone el sistema es incorrecto.

U3:	<i>Quería saber cuándo sale el primer tren el cuatro de Marzo por la tarde.</i>
U3:	(HORA-SALIDA) 0.92 FECHA-SALIDA: 04-05-???? 0.67 INTERVALO-HORA-SALIDA: 13.00-21.00 0.82
S3:	(INFORMACIÓN) 1.00 HORA-SALIDA: 17.00 1.00 FECHA-SALIDA-ACTUAL: 04-05-???? 0.67 INTERVALO-HORA-SALIDA-ACTUAL: 13.00-21.00 0.82 (NUEVA-CONSULTA) 1.00
S3:	<i>El cuatro de Mayo, el tren sale a las cinco de la tarde. ¿Desea algo más?</i>
U4:	<i>No era eso. Era Marzo. Quería trenes para el cuatro de Marzo.</i>
U4:	(NEGACIÓN) 0.85 (HORA-SALIDA) 0.78 FECHA-SALIDA: 04-03-???? 0.75
S4:	(INFORMACIÓN) 1.00 HORA-SALIDA: 15.30 1.00 FECHA-SALIDA-ACTUAL: 04-03-???? 0.75 (NUEVA-CONSULTA) 1.00
S4:	<i>El cuatro de Marzo, el tren sale a las tres y media de la tarde. ¿Desea algo más?</i>
U5:	<i>¿Hay alguno por la mañana?</i>
U5:	(HORA-SALIDA) 0.92 INTERVALO-HORA-SALIDA-ACTUAL: 7.00-13.00 0.82
S5:	(INFORMACIÓN) 1.00 HORA-SALIDA: 9.15 1.00 INTERVALO-HORA-SALIDA-ACTUAL: 7.00-13.00 0.82 (NUEVA-CONSULTA) 1.00
S5:	<i>El tren sale a las nueve y cuarto de la mañana. ¿Desea algo más?</i>

Figura 6.2: Fragmento de diálogo

La respuesta del sistema, S3, incluye una confirmación implícita de la fecha de salida, al tener una confianza inferior al umbral. Esto permite que el usuario detecte que la respuesta de horarios no corresponde a su petición y, por tanto, en U4, enmienda al sistema y reitera su petición. Esta vez, no se produce ningún error y el sistema responde correctamente a la

consulta del usuario. De nuevo, dada su medida de confianza, realiza la confirmación implícita de la fecha. Mas, como en el siguiente turno, U5, el usuario no rechaza esta fecha, el sistema puede asignarle una confianza más alta y omitir cualquier referencia a la misma en la siguiente respuesta, S5.

7. Simulación de los usuarios

7. Simulación de los usuarios

7.1 Introducción

Tal como se expuso en el apartado 2.6, varios grupos de investigación vienen considerando, y aplicando, técnicas de simulación de usuarios como una forma práctica de soslayar las dificultades que supone evaluar los sistemas de diálogo con usuarios reales, particularmente en las versiones preliminares de dichos sistemas.

Con estas técnicas de simulación se pretende: 1º) evaluar el funcionamiento del sistema de diálogo (o de algunos de sus módulos, como el gestor de diálogo), observando los efectos de introducir modificaciones (por ejemplo, diferentes estrategias de diálogo); 2º) mejorar el sistema de diálogo (por ejemplo, mediante la adaptación de los modelos de diálogo en función de los resultados de las simulaciones).

Estos mismos objetivos han determinado el desarrollo del módulo simulador de usuario que se presenta en esta sección. Información sobre este trabajo de investigación puede encontrarse también en (Torres et al., 2005b).

En esta sección, en primer lugar, se expone la arquitectura del sistema (resultado de integrar el sistema de diálogo con los módulos del simulador de usuario). A continuación, se presenta el algoritmo del simulador de usuario, que consiste en una versión modificada del gestor de diálogo visto en la Sección 5, pero adaptado para realizar el papel del usuario en los diálogos. Este módulo simulador de usuario tiene un comportamiento estocástico en la selección de transiciones a estados de sistema y un comportamiento fijado por reglas en la selección de transiciones a estados de usuario.

En el apartado 7.3 se explican las reglas de selección de las transiciones de usuario, fijadas conforme a una estrategia de diálogo colaborativa con el sistema. Después, en el apartado 7.4, se exponen los criterios seguidos en la generación de frames de usuario por parte del simulador y, en el apartado 7.5, se exponen las modificaciones introducidas en el gestor de diálogo del sistema, en particular el algoritmo de verificación del éxito de las simulaciones que permite la adaptación dinámica del modelo estocástico de diálogo en función del resultado de las simulaciones. A continuación, el funcionamiento del simulador de usuario se ilustra con un ejemplo de diálogo comentado en detalle. Por último, en el apartado 7.7, se describe el generador de respuestas del simulador, módulo encargado de traducir los frames de usuario a frases en lenguaje natural, mediante el uso de plantillas.

7.2 Arquitectura del sistema y algoritmo del simulador

Cuando el gestor de diálogo tenga que responder a los usuarios reales (ya sea usando modelos aprendidos a partir del corpus de BASURDE o del corpus de DIHANA), habrá cierta probabilidad de encontrarse en situaciones de diálogo que, aun siendo posibles en la tarea considerada, no se presentaron en los diálogos del corpus. En consecuencia, los modelos

(sobre todo, el de BASURDE) no proporcionarán un tratamiento adecuado a estas situaciones.

Como alternativa al aprendizaje de los modelos a partir de un corpus adquirido mediante la colaboración de usuarios reales, hemos considerado el aprendizaje por medio de la generación automática de diálogos. En este caso, la adquisición se lleva a cabo interactuando con módulos simuladores de los usuarios. Al aplicar esta técnica, se puede generar, rápidamente y a bajo coste, un corpus de diálogos simulados.

Además, si el comportamiento de los usuarios reales es adecuadamente modelado en los módulos simuladores, este corpus sintético puede usarse para adaptar los modelos iniciales, aprendido a partir del corpus de diálogos naturales. En este sentido, el componente del simulador que decide los actos de diálogo del usuario (el llamado *gestor de diálogo del usuario*):

- ❑ Se ha diseñado de modo que utiliza el mismo modelo de diálogo, MD, aprendido del corpus de BASURDE. Dado que MD es un modelo estadístico del comportamiento de usuarios reales en la adquisición de dicho corpus, la estrategia de diálogo que siga el simulador de usuario tendrá una parte estocástica, basada en las respuestas de tales usuarios.
- ❑ Además, para actuar de modo cooperativo con el sistema, el simulador utiliza un conjunto de reglas para elegir transiciones en el MD de acuerdo con los objetivos fijados en los escenarios objeto de simulación. Por tanto, la estrategia de diálogo del simulador está, en parte, determinada por reglas.

El simulador de usuario consta de dos módulos: un gestor de diálogo del usuario, GDU y un generador de respuestas del usuario, GRU:

- ❑ El GDU es el módulo simétrico del gestor de diálogo del sistema, GDS. Recibe los frames del sistema, lee y escribe su propio registro histórico, lee y transita en el mismo modelo de diálogo usado por el GDS, aplica un conjunto de reglas para elegir las transiciones de acuerdo a sus objetivos, y genera los frames de usuario.
- ❑ El GRU es el módulo simétrico del generador de respuestas del sistema; GRS. Recibe los frames de usuario y genera las correspondientes frases en lenguaje natural.

Por otra parte, el gestor de diálogo del sistema, GDS, se ha modificado de manera que pueda modificar su modelo de diálogo. Concretamente, cuando el GDS se usa en una adquisición sintética, puede detectar el éxito de los diálogos simulados y, si el éxito se verifica, añadir las transiciones elegidas en estos diálogos a su modelo (cambiando, por tanto, las probabilidades de las transiciones). En consecuencia, se puede adaptar el modelo de diálogo dinámicamente, al mismo tiempo que se realiza el proceso de generación de los diálogos.

En la Figura 7.1 se muestra el diagrama de bloques del sistema, extendido con los módulos de simulación del usuario.

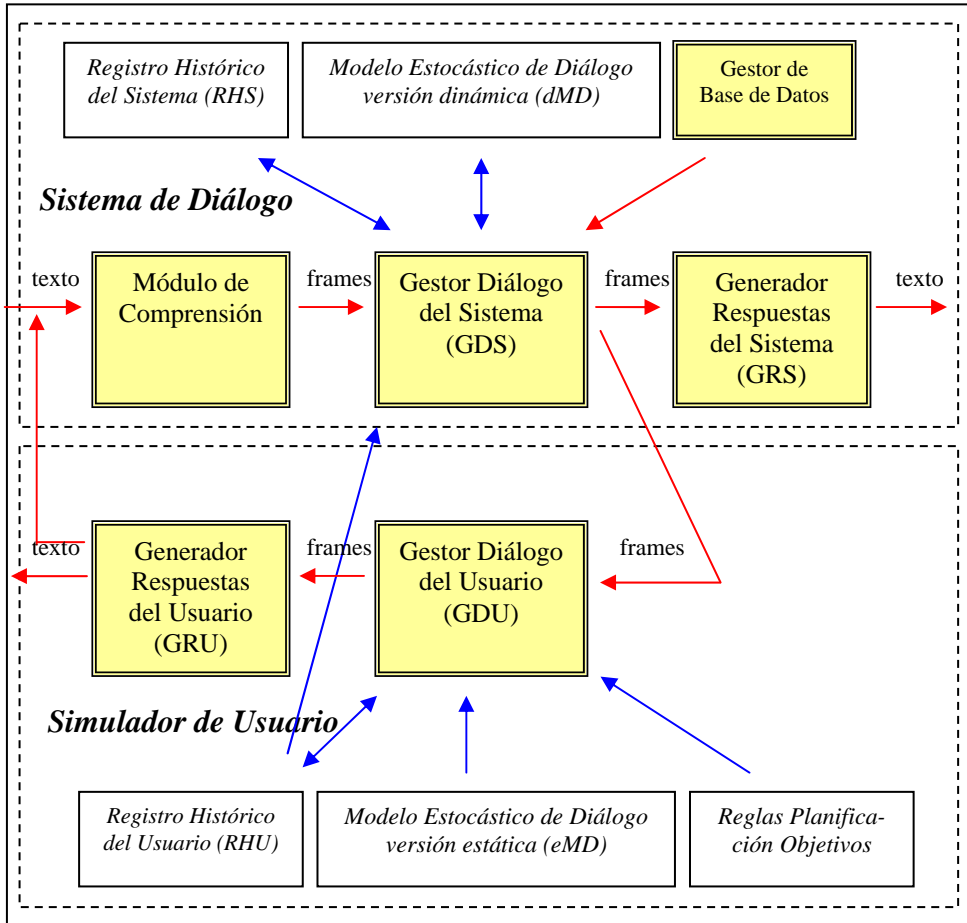


Figura 7.1: Diagrama de bloques del sistema de diálogo con simulador de usuario

En la Figura 7.2 se muestra el algoritmo del GDU o simulador de usuario. En la Figura 7.3 se muestra el nuevo algoritmo del GDS, que es una versión modificada del gestor de diálogo previamente descrito.

Al inicio de cada diálogo, el simulador de usuario:

- ❑ Lee los parámetros y objetivos del escenario simulado y almacena esta información en el RHU.
- ❑ Lee la versión estática del modelo, eMD, busca un estado de pregunta de, al menos, un objetivo del escenario, y genera el frame correspondiente a dicho estado.

En cada turno de diálogo, el simulador de usuario:

```

Inicializar(RHU);           // RHU = Registro Histórico del Usuario
Leer(eMD);                 // eMD = Modelo de Diálogo, estático
eMD.estado = PREGUNTA;    // estado inicial, Pregunta de un objetivo de escenario
frames-usuario = Adaptar(eMD.estado, RHU); // frames de usuario del estado inicial
Escribir(frames-usuario); // escritura de frames del usuario
REPETIR
    Leer (frames-sistema); // lectura de frames del sistema
                          // generalización semántica
    eMD.entrada = Adaptar(RHU, frames-sistema);           (1)
                          // transición a estado del sistema
    eMD.estado = Transitar(eMD.estado, eMD.entrada);      (2)
                          // actualización de RHU por turno de sistema
    RHU = Actualizar(RHU, frames-sistema);
                          // transición a estado del usuario
    eMD.estado = Transitar(eMD.estado, RHU, Reglas);      (3)
                          // generación de frames del usuario
    frames-usuario = Adaptar(eMD.estado, RHU);
    Escribir(frames-usuario); // escritura de frames del usuario
HASTA eMD.estado = CIERRE; // estado final, Cierre
Escribir(RHU);           // escritura Registro Histórico del Usuario

```

Figura 7.2: Algoritmo del simulador de usuario

- ❑ Lee los frames del sistema, generados por el GDS como respuesta a su anterior frame de usuario.
- ❑ Compara, mediante la generalización semántica, los frames del sistema recibidos y los actos de diálogo del sistema, posibles en el eMD (por disponer de transiciones en el modelo que lleven a estados de sistema correspondientes a tales actos de diálogo).
- ❑ Elige una transición, y así establece un nuevo estado del sistema en el eMD.
- ❑ Actualiza el RHU con los datos aportados por los frames del sistema.
- ❑ Transita a un nuevo estado de usuario en el eMD, conforme a un conjunto de reglas que establecen una estrategia de colaboración activa con el sistema.
- ❑ Genera los frames de usuario correspondientes al estado alcanzado en el eMD, incorporando los valores de atributos que sean precisos, si están disponibles en el RHU.

Al final del diálogo, el simulador de usuario escribe en disco una copia del contenido del RHU para que el GDS pueda consultarlo para verificar el éxito del diálogo.

```

Inicializar(RHS);           // RHS = Registro Histórico del Sistema
Leer(dMD);                 // dMD = Modelo de Diálogo, dinámico
dMD.estado = APERTURA;    // estado inicial, Apertura
REPETIR
  Leer(frames-usuario);    // lectura de frames de usuario
                          // generalización semántica
  dMD.entrada = Adaptar(RHS, frames-usuario);
                          // transición a estado de usuario
  dMD.estado = Transitar(dMD.estado, dMD.entrada);
                          // actualización de RHS por turno de usuario
  RHS = Actualizar(RHS, frames-usuario);
                          // transición a estado de sistema
  dMD.estado = Transitar(dMD.estado, RHS);
                          // actualización de RHS por turno de sistema
  RHS = Actualizar(RHS, dMD.estado, BD.salida);
                          // generación de frames de sistema
  frames-sistema = Adaptar(dMD.estado, RHS);
  Escribir(frames-sistema); // escritura de frames de sistema
HASTA dMD.estado = CIERRE; // estado final, Cierre
SI modo-de-simulación ENTONCES
  Leer(RHU);               // lectura Registro Histórico del Usuario
                          // verificar éxito por comparación de registros
  éxito = Comparar(RHU, RHS);
                          // modificación dinámica del modelo de diálogo
  SI éxito ENTONCES Escribir(dMD, transiciones);

```

Figura 7.3: Algoritmo del gestor de diálogo

7.3 Reglas de selección de transiciones de usuario

Como se puede observar, los algoritmos del GDS y el GDU son muy parecidos. La diferencia principal se encuentra en el procedimiento de elegir transiciones en su propio turno. El GDU procesa el turno del sistema usando MD como un modelo de bigramas y aplicando la generalización semántica, del mismo modo en que lo hace el GDS (véanse las líneas 1 y 2 en la Figura 7.2).

Sin embargo, cuando el GDU decide su respuesta (en su propio turno, o turno de usuario), usa el MD como una lista de posibles nuevos estados, sin tener en cuenta la existencia de transiciones que lleven a ellos. Entonces, aplica un conjunto de reglas para buscar el mejor nuevo estado del usuario, de acuerdo con el contenido actual del RHU y dada una estrategia de colaboración que busca satisfacer los objetivos del escenario simulado (línea 3, Figura 7.2).

En la Figura 7.4 se muestran las principales reglas para elegir un estado de usuario. La primera regla establece que, después de una respuesta del sistema, el GDU verifica la existencia de objetivos del escenario cuyo valor todavía desconozca y, dependiendo de esto,

<p>SI estado-sistema = <i>Respuesta</i> <i>Apertura</i> <i>Nueva-Consulta</i> ENTONCES buscar nuevos objetivos en RHU SI $\exists!$ atributo / es-objetivo & valor-desconocido ENTONCES (buscar & transitar) estado-usuario = <i>Pregunta</i> de atributo objetivo SINO (buscar & transitar) estado-usuario = <i>Cierre</i></p> <p>SI estado-sistema = <i>Pregunta</i> ENTONCES buscar atributo(s) preguntado(s) en RHU SI atributo(s) / valor-conocido ENTONCES (buscar & transitar) estado-usuario = <i>Respuesta</i> de atributo(s) SINO (buscar & transitar) estado-usuario = <i>Pregunta</i> de atributo objetivo</p> <p>SI estado-sistema = <i>Confirmación</i> ENTONCES buscar atributo(s) preguntado(s) en RHU SI atributo(s) / valor-conocido ENTONCES SI atributo-valor(RHU) = atributo-valor(frames) ENTONCES (buscar & transitar) estado-usuario = <i>Afirmación</i> de atributo(s) SINO (buscar & transitar) estado-usuario = <i>Negación</i> de atributo(s) SINO (buscar & transitar) estado-usuario = <i>Pregunta</i> de atributo objetivo</p>
--

Figura 7.4: Reglas aplicadas para elegir estado de usuario en el simulador

transita a un estado de pregunta, de ese objetivo aún no satisfecho, o al estado de cierre del diálogo.

La segunda regla establece que, después de una pregunta sobre atributos cuyo valor ignora el sistema, el GDU busca en el RHU los valores de los atributos pedidos y, si los encuentra (porque en la lectura del escenario simulado se les asignó valores), transita a un estado de respuesta de los mismos. En caso contrario, la transición será a algún estado de pregunta de un atributo objetivo. En este último caso, hay todavía cierta probabilidad de satisfacer parte de la pregunta del sistema mediante el aporte espontáneo de datos disponibles en el RHU.

La tercera regla establece que, después de una petición de confirmación de valores de atributos, el GDU verifica si conoce los valores de los atributos consultados, los compara con los valores que supone el sistema y, dependiendo de su igualdad, transita a un estado de afirmación o a un estado de negación de tales atributos. De nuevo, en caso de ignorar el valor de los atributos consultados, transita a algún estado de pregunta de un atributo objetivo.

7.4 Generación de los frames de usuario

Los frames que suministra el GDU a su generador de respuestas, GRU, quedan determinados fundamentalmente por el estado alcanzado en el modelo. Sin embargo, la información almacenada en su RHU también juega un papel relevante y permite incrementar la variabilidad de las posibles respuestas del simulador.

Dados los escenarios considerados, el estado de usuario al que más frecuentemente transita en el GDU en el primer turno de diálogo es el de consulta de horarios, (U:PREGUNTA:HORA-SALIDA). Esto sólo determina que se genere un frame (HORA-SALIDA), pero no qué atributos le acompañarán. Se pueden considerar varias estrategias:

- ❑ No aportar ningún atributo. El simulador los facilitaría, en los siguientes turnos, conforme el sistema, mediante preguntas y confirmaciones, se los solicitara. Esta estrategia representaría a un usuario nada colaborador, casi irreal (alguien que, en su primera intervención, dijera, tan sólo: *¿a qué hora sale?*, sin dar más información). Obviamente, esta estrategia no se considera.
- ❑ Aportar todos los atributos disponibles en el escenario. Representa el extremo contrario al anterior. El simulador proporcionaría en el turno inicial toda la información, e incluso tendería a repetirla en los turnos siguientes. Igualmente, parece un comportamiento poco real.
- ❑ Incluir en cada turno sólo un subconjunto de los atributos disponibles en el escenario. Se pueden considerar diferentes criterios para seleccionarlos: de modo aleatorio, basado en probabilidades (confianzas), basado en estado de confirmación o rechazo, etc. Este planteamiento resulta más razonable y, además, facilita la variabilidad de las posibles respuestas.

Hay turnos de usuario donde, estrictamente, no es obligatorio facilitar ningún valor de atributo, como en el ejemplo anterior, en el estado (U:PREGUNTA:HORA-SALIDA). También hay turnos donde es preciso facilitar algunos valores, como sería el caso del estado (U:RESPUESTA:DESTINO,ORIGEN), donde el correspondiente frame tiene que incluir los valores de las ciudades origen y destino del viaje. Tanto en un tipo de turnos como en el otro, los frames pueden incluir otros atributos. Llamamos *aporte espontáneo de información* a la presencia de estos atributos opcionales en los frames de usuario.

Se han establecido diferentes criterios para implementar este aporte espontáneo de datos en función de que se trate de la intervención inicial del simulador o de que se trate de cualquiera de los turnos siguientes.

En el caso del primer turno del diálogo, la inclusión de atributos en los frames está condicionada a la inicialización del RHU. En este registro, para cada atributo presente en el escenario simulado, además de guardarse su valor, hay que asignar una probabilidad (que se puede interpretar como su confianza), un estado de confirmación y una edad (instante de modificación del valor). Un atributo con “edad baja” significa que ha sido modificado recientemente y, por tanto, es un candidato más probable a ser objeto de confirmación implícita (en el GDS) o a ser objeto del aporte espontáneo de información (en el GDU).

La regla para incluir un atributo como aporte espontáneo en un frame (siempre que sea compatible con el atributo) es la siguiente: incluir si la edad del atributo en el RHU es menor o igual a 2.

Parece razonable suponer que los usuarios reales incluirán, en su primer turno de diálogo, los atributos que fijan el trayecto (origen, destino y fecha) con más frecuencia que otros atributos (tipo de tren, intervalo horario, etc). Por ello, en la inicialización del RHU, se asigna: a) una probabilidad aleatoria dentro del rango [0.3, 0.8] y una edad aleatoria dentro del rango [1, 3] a los atributos ORIGEN, DESTINO y FECHA-SALIDA; b) una probabilidad

aleatoria dentro del rango [0.5, 1.0] y una edad aleatoria dentro del rango [1, 4] al resto de atributos que figuren en el escenario simulado.

Este criterio facilita que los atributos ORIGEN, DESTINO y FECHA-SALIDA se incorporen a los frames generados en el turno inicial del simulador de usuario. Es decir, estos atributos tendrán más probabilidad de aparecer, pero también puede suceder que estén ausentes alguno o todos, o que aparezcan otros atributos como TIPO-TREN o INTERVALO-HORA-SALIDA.

En el caso de los siguientes turnos del diálogo, se aplica un criterio complementario. En el RHU, se modifican los valores de probabilidad y de edad de varios atributos. Los atributos que, al inicio, recibieron una edad aleatoria dentro del rango [1, 4] tienen, ahora, más probabilidad de que su edad se cambie al rango [1, 2] y, en consecuencia, de que sean incorporados a los frames de este turno. Para cada diálogo, se elige aleatoriamente el umbral que determina qué atributos se ven afectados por este cambio. Este criterio complementa al seguido en el turno inicial y favorece a otros atributos como TIPO-TREN o INTERVALO-HORA-SALIDA que tenían menos probabilidad de figurar en aquel turno.

```

SI número-de-turnos > T-max      /* verificación de la duración del diálogo */
ENTONCES Escribir-Error("excedida duración límite")
                                  /* verificación de parámetros de la consulta */
SI ∃! atributo / es-parámetro-de-consulta
    & atributo-valor(RHU) != atributo-valor(RHS)
ENTONCES Escribir-Error("parámetros de la consulta diferentes")
                                  /* verificación de objetivos de la consulta */
SI ∃! atributo / es-objetivo-de-consulta
    & atributo-valor(RHU) != atributo-valor(RHS)
ENTONCES Escribir-Error("objetivos de la consulta diferentes")
SI ∀ atributo / atributo ∈ escenario      /* validación del diálogo */
    & atributo-valor(RHU) = atributo-valor(RHS)
ENTONCES Escribir-Éxito("final del diálogo con éxito")
        Escribir(dDM, transiciones)      /* adición de transiciones al modelo */

```

Figura 7.5: Algoritmo de evaluación del éxito de las simulaciones

7.5 Modificaciones en el gestor de diálogo

En cuanto al nuevo algoritmo del GDS, es preciso destacar los dos cambios principales, introducidos para que funcione integrado con el simulador de usuario y pueda adaptar el modelo de diálogo:

- En cada turno de diálogo, las transiciones elegidas en su modelo de diálogo, dMD, se añaden, temporalmente, al propio modelo. Para ello, se reajusta, al alza, las probabilidades de las transiciones elegidas y, a la baja, las probabilidades de las otras transiciones, disponibles desde el estado actual, pero no elegidas.

- Al final de cada diálogo, se lee el RHU y se comparan los dos registros históricos (RHU y RHS) para determinar el éxito de la simulación. Todas las transiciones que se añadieron temporalmente al modelo, se consolidan (se añaden permanentemente) en el caso de conclusión satisfactoria del diálogo. En cambio, cuando el diálogo simulado resulta fallido, las transiciones añadidas se eliminan, restaurando el modelo a tal como era al inicio del diálogo.

En la Figura 7.5 se muestra el algoritmo usado para evaluar automáticamente el éxito de las simulaciones.

RHU	
edad = 2; conf = 1; prob = 0.61; valor = Toledo	[ORIGEN]
edad = 1; conf = 1; prob = 0.67; valor = Salamanca	[DESTINO]
edad = 2; conf = 1; prob = 0.80; valor = 04-11-????	[FECHA-SALIDA]
edad = 1; conf = 0; prob = 0.41; valor = ???	[HORA-SALIDA]
edad = 4; conf = 0; prob = 0.10; valor = ???	[HORA-LLEGADA]
edad = 4; conf = 1; prob = 0.78; valor = 05.00-13.00	[INTERVALO-HORA-SALIDA]

MDU	
estado-sistema = (S:APERTURA:NIL)	
estado-usuario = (U:PREGUNTA:HORA-SALIDA)	

U0: <i>A qué hora sale desde Toledo a Salamanca el día cuatro de Noviembre.</i>	

Figura 7.6: Ejemplo de diálogo, 1º turno del simulador de usuario

7.6 Ejemplo de diálogo

A continuación, se muestra y comenta un ejemplo de diálogo obtenido mediante simulación de usuario y donde, además, se han introducido errores de sustitución en algunos atributos. En las figuras que ilustran el ejemplo, se usa la abreviatura MDU (modelo de diálogo del usuario) para referirse al eMD, el modelo usado por el GDU, y la abreviatura MDS (modelo de diálogo del sistema) para referirse al dMD, el modelo usado por el GDS.

El escenario simulado consiste en obtener horarios de salida y llegada para un viaje de Toledo a Salamanca el cuatro de Noviembre por la mañana. Por tanto, el simulador de usuario lee los valores del escenario y los almacena en su RHU, tal como se indica en la Figura 7.6, que reproduce el primer turno del usuario.

Conforme a los criterios de inicialización del RHU, a los atributos ORIGEN, DESTINO y FECHA-SALIDA se les asigna una probabilidad en el rango [0.3, 0.8] y una edad en el rango [1, 3], mientras que el resto de atributos reciben una probabilidad en el rango [0.5, 1.0] y una edad en el rango [1, 4]. En el modelo, MDU, el simulador transita al estado (U:PREGUNTA:HORA-SALIDA), correspondiente a uno de los objetivos del escenario. Esto determina la generación de un frame (HORA-SALIDA). Al mismo se incorporan, como

```

#frase simulada: A qué hora sale desde Mallorca a Salamanca el día veintinueve de
Febrero.
(HORA-SALIDA) 0.41
  ORIGEN: Mallorca 0.75
  DESTINO: Salamanca 0.88
  FECHA-SALIDA: 29-02-???? 0.58
-----
RHS
edad = 1; conf = 0; prob = 0.75; valor = Mallorca   [ORIGEN]
edad = 1; conf = 0; prob = 0.88; valor = Salamanca [DESTINO]
edad = 1; conf = 0; prob = 0.58; valor = 29-02-???? [FECHA-SALIDA]
edad = 1; conf = 0; prob = 0.00; valor = ???       [HORA-SALIDA]
-----
MDS
estado-usuario = (U:PREGUNTA:HORA-SALIDA)
-----
transiciones podadas con prob>0.1
prob=0.1871; (S:Confirmación:Hora-Salida)(S:Espera:Nil)(S:Respuesta:Hora-
Salida)(S:Nueva-Consulta:Nil)
prob=0.1871; (S:Confirmación:Destino,Origen)
prob=0.1259; (S:Respuesta:Hora-Salida)(S:Nueva-Consulta:Nil)
prob=0.1259; (S:Confirmación:Destino,Fecha-Salida,Origen)
prob=0.1583; (S:Espera:Nil)(S:Respuesta:Hora-Salida)(S:Nueva-Consulta:Nil)
-----
transiciones permitidas con prob>0.01
prob=0.0683 → 0.1741 ; (S:Confirmación:Fecha-Salida)
prob=0.0252 → 0.0176 ; (S:Confirmación:Hora-Salida)(S:Espera:Nil)
prob=0.0036 → 0.1288 ; (S:Pregunta:Fecha-Salida)
-----
máximo = 0.1741; umbral_aleat = 0.0435
estado-sistema = (S:CONFIRMACIÓN:FECHA-SALIDA)
-----
RHS
edad = 2; conf =-2; prob = 0.75; valor = Mallorca   [ORIGEN]
edad = 2; conf =-2; prob = 0.88; valor = Salamanca [DESTINO]
edad = 2; conf =-1; prob = 0.71; valor = 29-02-???? [FECHA-SALIDA]
edad = 2; conf = 0; prob = 0.00; valor = ???       [HORA-SALIDA]
-----
(CONFIRMACIÓN) 1.00
  FECHA-SALIDA: 29-02-???? 0.71
  ORIGEN-ACTUAL: Mallorca 0.75
  DESTINO-ACTUAL: Salamanca 0.88
S0: Desde Mallorca, ¿quiere viajar el veintinueve de Febrero?

```

Figura 7.7: Ejemplo de diálogo, 1º turno del sistema

atributos tipo ACTUAL, los atributos cuya edad sea menor o igual a 2, que, en este caso, son justamente los atributos ORIGEN, DESTINO y FECHA-SALIDA.

A continuación, en el módulo de comprensión, se han simulado dos errores: a) la ciudad de origen se ha cambiado por Mallorca; b) la fecha de salida se ha cambiado al 29 de Febrero. Estos errores se propagan al gestor de diálogo, cuyo primer turno de diálogo se reproduce en la Figura 7.7.

El gestor de diálogo almacena, en su RHS, los valores y confianzas de los atributos así como la marca de consulta de horarios de salida. Además, el nuevo estado de usuario en el MDS es (U:PREGUNTA:HORA-SALIDA).

En la selección del siguiente estado de sistema se han indicado las principales transiciones podadas y permitidas. Al respecto de las transiciones podadas, se puede destacar que: a) se podan transiciones a estados de respuesta, dada la baja confianza del atributo FECHA-SALIDA; b) se podan transiciones a estados de confirmación que incluyen el atributo DESTINO, dada la alta confianza del mismo. En cuanto a las transiciones permitidas, se aprecia que: a) las dos transiciones relativas a preguntar o confirmar la FECHA-SALIDA incrementan su probabilidad, dada la baja confianza del atributo; b) la otra posibilidad, una confirmación de la consulta de horarios, es penalizada.

Para elegir una de las transiciones, se ha generado un umbral aleatorio, relativo a la probabilidad máxima de las transiciones permitidas (en este caso, umbral = $0.0435 = (1/4) \cdot 0.1741$). La probabilidad de la transición a confirmar la consulta de horarios es inferior a este umbral y se descarta. Entre las restantes, se ha elegido, aleatoriamente, como nuevo estado del sistema (S:CONFIRMACIÓN:FECHA-SALIDA).

En el RHS, el atributo FECHA-SALIDA queda marcado como objeto de confirmación en curso y se incrementa su probabilidad, en un valor aleatorio en el rango $[0.1, 0.4]$, valor de confianza que se consolidará si en el siguiente turno el usuario no rechaza el valor del atri-

<p>MDU estado-sistema = (S:CONFIRMACIÓN:FECHA-SALIDA) estado-usuario = (U:NEGACIÓN:FECHA-SALIDA)(U:RESPUESTA:FECHA-SALIDA) -----</p> <p>RHU edad = 2; conf = 1; prob = 0.61; valor = Toledo [ORIGEN] edad = 3; conf = 1; prob = 0.95; valor = Salamanca [DESTINO] edad = 2; conf = 1; prob = 0.80; valor = 04-11-???? [FECHA-SALIDA] edad = 3; conf = 0; prob = 0.41; valor = ??? [HORA-SALIDA] edad = 6; conf = 0; prob = 0.10; valor = ??? [HORA-LLEGADA] edad = 2; conf = 1; prob = 0.78; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA] -----</p> <p>(NEGACIÓN) 1.00 (INFORMACIÓN) 1.00 FECHA-SALIDA: 04-11-???? 0.80 ORIGEN-ACTUAL: Toledo 0.61 INTERVALO-HORA-SALIDA-ACTUAL: 05.00-13.00 0.78 U1: <i>No, quiero viajar el cuatro de Noviembre desde Toledo salida por la mañana.</i></p>

Figura 7.8: Ejemplo de diálogo, 2º turno del simulador de usuario

```

(NEGACIÓN) 0.88
( ) 0.86
  ORIGEN: Toledo 0.86
  FECHA-SALIDA: 04-11-2005 0.63
  INTERVALO-HORA-SALIDA: 05.00-13.00 0.44
-----
RHS
edad = 1; conf = 0; prob = 0.86; valor = Toledo           [ORIGEN]
edad = 3; conf = 0; prob = 0.88; valor = Salamanca [DESTINO]
edad = 1; conf = 0; prob = 0.63; valor = 04-11-2005 [FECHA-SALIDA]
edad = 3; conf = 0; prob = 0.00; valor = ???           [HORA-SALIDA]
edad = 1; conf = 0; prob = 0.44; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]
-----
MDS
Generalización semántica
  (U:Negacion:Fecha_salida)(U:Respuesta:Fecha_salida,Intervalo_hora_salida,
  Origen)
estado-usuario = (U:NEGACIÓN:FECHA-SALIDA)
-----
transiciones permitidas con prob>0.1
prob=0.8571 → 0.7098 ; (S:Pregunta:Fecha_salida)
prob=0.1429 → 0.2100 ; (S:No_entendido:Nil) (S:Pregunta:Fecha_salida)
-----
máximo = 0.7098; umbral_aleat = 0.1417
estado-sistema = (S:PREGUNTA:FECHA-SALIDA)
-----
RHS
edad = 2; conf = -2; prob = 0.86; valor = Toledo           [ORIGEN]
edad = 4; conf = 0; prob = 0.88; valor = Salamanca [DESTINO]
edad = 2; conf = -1; prob = 0.74; valor = 04-11-2005 [FECHA-SALIDA]
edad = 4; conf = 0; prob = 0.00; valor = ???           [HORA-SALIDA]
edad = 2; conf = -2; prob = 0.44; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]
-----
(SOLICITUD) 1.00
  FECHA-SALIDA: 04-11-2005 0.74
  ORIGEN-ACTUAL: Toledo 0.86
  INTERVALO-HORA-SALIDA-ACTUAL: 05.00-13.00 0.44
S1: Salida por la mañana, ¿qué día quiere salir?

```

Figura 7.9: Ejemplo de diálogo, 2º turno del sistema

buto. En el frame de confirmación generado, se incorporan los atributos ORIGEN y DESTINO de tipo ACTUAL. De estos atributos, sólo la ciudad de origen aparece en la respuesta en lenguaje natural del sistema, al ser el atributo que tiene una confianza menor al umbral 0.8 fijado en el generador de respuestas.

Como se observa en la Figura 7.8, que reproduce el siguiente turno del simulador de usuario, la intervención del sistema es adecuadamente reconocida, con una transición, en el MDU, al mismo estado (S:CONFIRMACIÓN:FECHA-SALIDA). A continuación, el simulador

elige transición a estado de usuario aplicando la tercera regla de la Figura 7.4: como el dato propuesto por el sistema es incorrecto, la respuesta será rechazarlo y aportar de nuevo el dato correcto. Esta acción, en términos de modelo de diálogo, implica transitar al estado (U:NEGACIÓN:FECHA-SALIDA) (U:RESPUESTA:FECHA-SALIDA).

En el RHU, se modifican los valores de edad de varios atributos, según el criterio fijado para todos los turnos del simulador, excepto el turno inicial. Por ello, en el frame (INFORMACIÓN), correspondiente al acto de diálogo de respuesta de la fecha de salida, se incluyen también, como atributos tipo ACTUAL, la ciudad de origen y el intervalo horario. La inclusión de los atributos ORIGEN y INTERVALO-HORA-SALIDA se produce como un aporte espontáneo de datos. En el caso del atributo ORIGEN, su incorporación permite que, en el mismo turno, el simulador corrija el error presente en la confirmación implícita del sistema (además de corregir el error en la confirmación explícita de la fecha).

En el siguiente turno del sistema (véase la Figura 7.9), se almacenan los valores correctos de los tres atributos en el RHS, dado que no se introducen errores. Así pues, la confirmación mixta del turno anterior ha permitido la corrección por parte del simulador de usuario. Como el simulador no ha generado el frame (AFIRMACIÓN), en RHS el atributo FECHA-SALIDA queda, como al principio, sin confirmar.

En este caso, al recibirse dos frames de usuario, resulta de interés el proceso de generalización semántica. El frame (NEGACIÓN) se equipara al correspondiente acto de diálogo de negación del día de salida. El frame incompleto se corresponde con un acto de diálogo de respuesta, con los atributos presentes en el frame. Así, la cadena inicial para buscar transiciones en el MDS es (U:NEGACIÓN:FECHA-SALIDA) (U:RESPUESTA:FECHA-SALIDA,INTERVALO-HORA-SALIDA,ORIGEN).

El modelo se actualiza mediante fragmentación aleatoria y transita al estado (U:NEGACIÓN:FECHA-SALIDA). Aunque el valor correcto de la fecha de salida ha quedado almacenado en el RHS, las dos únicas transiciones permitidas a estados de sistema consisten en preguntar la fecha de salida. Se elige el estado (S:PREGUNTA:FECHA-SALIDA).

<p>MDU estado-sistema = (S:PREGUNTA:FECHA-SALIDA) estado-usuario = (U:RESPUESTA:FECHA-SALIDA)</p> <p>-----</p> <p>RHU edad = 6; conf = 1; prob = 0.95; valor = Toledo [ORIGEN] edad = 5; conf = 1; prob = 0.95; valor = Salamanca [DESTINO] edad = 6; conf = 1; prob = 0.80; valor = 04-11-???? [FECHA-SALIDA] edad = 5; conf = 0; prob = 0.41; valor = ??? [HORA-SALIDA] edad = 8; conf = 0; prob = 0.10; valor = ??? [HORA-LLEGADA] edad = 6; conf = 1; prob = 0.78; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]</p> <p>-----</p> <p>(INFORMACIÓN) 1.00 FECHA-SALIDA: 04-11-???? 0.80 U2: <i>Quiero viajar el cuatro de Noviembre.</i></p>

Figura 7.10: Ejemplo de diálogo, 3º turno del simulador de usuario

En el siguiente turno del usuario (véase la Figura 7.10), se facilita de nuevo el dato solicitado por el sistema. A continuación (véase la Figura 7.11, en páginas siguientes), el sistema recibe la respuesta del simulador y, aplicando una trivial generalización semántica, transita a (U:RESPUESTA:FECHA-SALIDA).

Entre las transiciones podadas, se encuentra la que llevaría a responder con horarios del viaje de vuelta (podada porque el viaje actual está identificado como de ida en el RHS) y la de respuesta de horarios de llegada (podada porque tal atributo aún no está marcado como preguntado en el RHS). De nuevo, para elegir una de las transiciones, se ha generado un umbral aleatorio, relativo a la probabilidad máxima de las transiciones permitidas (en este caso, $\text{umbral} = 0.2140 = (1/4) 0.8559$). La probabilidad de la transición a confirmar la fecha de salida es inferior a este umbral y se poda también en esta última criba. Por tanto, sólo queda como transición disponible la más probable, y el nuevo estado del sistema es (S:CONFIRMACIÓN:HORA-SALIDA)(S:ESPERA:NIL)(S:RESPUESTA:HORA-SALIDA)(S:NUEVA-CONSULTA:NIL).

Tras esto, se generan los correspondientes frames y frases en español, informando al usuario (a su simulador) del horario de salida del primer tren.

En consecuencia, en el cuarto turno del simulador (véase la Figura 7.12), el horario facilitado se almacena en el RHU, quedando satisfecho uno de los objetivos del escenario. La transición, en MDU, a nuevo estado de sistema se consigue, mediante el método de generalización semántica llamado concatenación condicional, al contemplar únicamente el último frame del sistema. A continuación la transición de (S:NUEVA-CONSULTA:NIL) a (U:PREGUNTA:HORA-LLEGADA) se explica por la aplicación de la primera regla de la Figura 7.4: queda un objetivo por satisfacer, la obtención del horario de llegada, y se busca un estado de pregunta del mismo.

En el siguiente turno del sistema (véase la Figura 7.13), se transita al estado de respuesta del horario de llegada. Y, finalmente, en el siguiente turno del simulador (véase la Figura 7.14), una vez satisfecho este último objetivo del escenario, y por aplicación de la misma primera regla de la Figura 7.4, se transita al estado de cierre del diálogo.

Este ejemplo corresponde a un diálogo concluido con éxito, dada la igualdad de contenido de los dos registros, RHS y RHU, al final del diálogo.

() 0.86

FECHA-SALIDA: 04-11-2005 0.63

RHS

edad = 2; conf = 0; prob = 0.95; valor = Toledo [ORIGEN]
 edad = 5; conf = 0; prob = 0.88; valor = Salamanca [DESTINO]
 edad = 3; conf = 0; prob = 0.63; valor = 04-11-2005 [FECHA-SALIDA]
 edad = 5; conf = 0; prob = 0.00; valor = ??? [HORA-SALIDA]
 edad = 3; conf = 0; prob = 0.62; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]

MDS

estado-usuario = (U:RESPUESTA:FECHA-SALIDA)

transiciones podadas con prob>0.01

prob=0.0588;

(S:Confirmacion:Hora_salida_v)(S:Espera:Nil)(S:Respuesta:Hora_salida_v)(S:Nueva_consulta:Nil)

prob=0.0294;

(S:Confirmacion:Hora_llegada)(S:Espera:Nil)(S:Respuesta:Hora_llegada)(S:Nueva_consulta:Nil)

transiciones permitidas con prob>0.01

prob=0.7941 → 0.8559 ;

(S:Confirmacion:Hora_salida)(S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)

prob=0.1176 → 0.1923 ; (S:Confirmacion:Dia_salida)

máximo = 0.8559; umbral_aleat = 0.2140

estado-sistema = (S:CONFIRMACIÓN:HORA-

SALIDA)(S:ESPERA:NIL)(S:RESPUESTA:HORA-SALIDA)(S:NUEVA-CONSULTA:NIL)

RHS

edad = 3; conf = 0; prob = 0.95; valor = Toledo [ORIGEN]
 edad = 6; conf = 0; prob = 0.88; valor = Salamanca [DESTINO]
 edad = 0; conf = 0; prob = 0.99; valor = 05 [NÚMERO-TRENES]
 edad = 4; conf = 0; prob = 0.63; valor = 04-11-2005 [FECHA-SALIDA]
 edad = 0; conf = 0; prob = 0.99; valor = 06.31 [HORA-SALIDA]
 edad = 4; conf = 0; prob = 0.62; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]

(CONFIRMACIÓN) 1.00

HORA-SALIDA: ??? 0.00

(ESPERA) 1.00

(INFORMACIÓN) 1.00

NÚMERO-TRENES: 05 0.99

HORA-SALIDA: 06.31 0.99

(NUEVA-CONSULTA) 1.00

S2: *Hay 5 trenes, y le informo del primero. Le consulto la hora de salida. Un momento por favor. El tren sale a las siete menos veintinueve de la mañana. ¿Desea algo más?*

Figura 7.11: Ejemplo de diálogo, 3º turno del sistema

MDU
 Generalización semántica
 (S:Confirmacion:Hora_salida)(S:Espera:Nil)(S:Respuesta:Hora_salida,Numero_trenes)
)(S:Nueva_consulta:Nil)
 modelo actualizado mediante concatenación incondicional con (S:Nueva_consulta:Nil)
 estado-sistema = (S:NUEVA-CONSULTA:NIL)
 estado-usuario = (U:PREGUNTA:HORA-LLEGADA)

RHU
 edad = 8; conf = 1; prob = 0.95; valor = Toledo [ORIGEN]
 edad = 7; conf = 1; prob = 0.95; valor = Salamanca [DESTINO]
 edad = 2; conf = 0; prob = 0.99; valor = 05 [NÚMERO-TRENES]
 edad = 7; conf = 1; prob = 0.80; valor = 04-11-???? [FECHA-SALIDA]
 edad = 2; conf = 0; prob = 0.99; valor = 06.31 [HORA-SALIDA]
 edad = 10; conf = 0; prob = 0.10; valor = ??? [HORA-LLEGADA]
 edad = 8; conf = 1; prob = 0.78; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]

(SOLICITUD) 1.00
 HORA-LLEGADA: ??? 0.10
 NÚMERO-TRENES-ACTUAL: 05 0.99
 HORA-ACTUAL: 06.31 0.99
U3: *A qué hora llega.*

Figura 7.12: Ejemplo de diálogo, 4º turno del simulador de usuario

MDU
 estado-sistema = (S:ESPERA:NIL)(S:RESPUESTA:HORA-LLEGADA)(S:NUEVA-CONSULTA:NIL)
 estado-usuario = (U:CIERRE:NIL)

RHU
 edad = 10; conf = 1; prob = 0.95; valor = Toledo [ORIGEN]
 edad = 9; conf = 1; prob = 0.95; valor = Salamanca [DESTINO]
 edad = 4; conf = 0; prob = 0.99; valor = 05 [NÚMERO-TRENES]
 edad = 9; conf = 1; prob = 0.80; valor = 04-11-???? [FECHA-SALIDA]
 edad = 4; conf = 0; prob = 0.99; valor = 06.31 [HORA-SALIDA]
 edad = 2; conf = 0; prob = 0.99; valor = 07.31 [HORA-LLEGADA]
 edad = 10; conf = 1; prob = 0.78; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]

(CIERRE) 1.00
U4: *Nada más muchas gracias.*

Figura 7.14: Ejemplo de diálogo, 5º turno del simulador de usuario

(HORA-LLEGADA) 0.49

RHS

edad = 4; conf = 0; prob = 0.95; valor = Toledo [ORIGEN]
 edad = 7; conf = 0; prob = 0.88; valor = Salamanca [DESTINO]
 edad = 1; conf = 0; prob = 0.99; valor = 05 [NÚMERO-TRENES]
 edad = 5; conf = 0; prob = 0.63; valor = 04-11-2005 [FECHA-SALIDA]
 edad = 1; conf = 0; prob = 0.99; valor = 06.31 [HORA-SALIDA]
 edad = 1; conf = 0; prob = 0.00; valor = ??? [HORA-LLEGADA]
 edad = 5; conf = 0; prob = 0.62; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]

MDS

estado-usuario = (U:PREGUNTA:HORA-LLEGADA)

 transiciones podadas con prob>0.01
 prob=0.0179; (S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)

 transiciones permitidas con prob>0.01
 prob=0.4643 → 0.6250 ; (S:Espera:Nil)(S:Respuesta:Hora_llegada)(S:Nueva_consulta:Nil)
 prob=0.4464 → 0.6125 ; (S:Respuesta:Hora_llegada)(S:Nueva_consulta:Nil)
 prob=0.0357 → 0.3250 ;
 (S:Confirmacion:Hora_llegada)(S:Espera:Nil)(S:Respuesta:Hora_llegada)(S:Nueva_consulta:Nil
)
 prob=0.0179 → 0.0179 ; (S:No_entendido:Nil)(S:Nueva_consulta:Nil)
 prob=0.0179 → 0.0125 ; (S:Confirmacion:Hora_llegada)(S:Espera:Nil)

 máximo = 0.6250; umbral_aleat = 0.1250
 estado-sistema = (S:ESPERA:NIL)(S:RESPUESTA:HORA-LLEGADA)(S:NUEVA-CONSULTA:NIL)

RHS

edad = 5; conf = 0; prob = 0.95; valor = Toledo [ORIGEN]
 edad = 8; conf = 0; prob = 0.88; valor = Salamanca [DESTINO]
 edad = 2; conf = 0; prob = 0.99; valor = 05 [NÚMERO-TRENES]
 edad = 6; conf = 0; prob = 0.63; valor = 04-11-2005 [FECHA-SALIDA]
 edad = 2; conf = 0; prob = 0.99; valor = 06.31 [HORA-SALIDA]
 edad = 0; conf = 0; prob = 0.99; valor = 07.31 [HORA-LLEGADA]
 edad = 6; conf = 0; prob = 0.62; valor = 05.00-13.00 [INTERVALO-HORA-SALIDA]

 (ESPERA) 1.00

(INFORMACIÓN) 1.00

HORA-LLEGADA: 07.31 0.99

NÚMERO-TRENES-ACTUAL: 05 0.99

HORA-ACTUAL: 06.31 0.99

(NUEVA-CONSULTA) 1.00

S3: *Un momento por favor. El tren llega a las ocho menos veintinueve de la mañana.
 ¿Desea algo más?*

Figura 7.13: Ejemplo de diálogo, 4º turno del sistema

7.7 Generador de respuestas del simulador de usuario

El módulo generador de respuestas del simulador de usuario (GRU) traduce las representaciones semánticas de los turnos de usuario (la salida del módulo GDU, gestor de diálogo de usuario) a frases en español usando plantillas y reglas de combinación de estas plantillas con los valores de los atributos.

Se trata, prácticamente, del mismo generador de respuestas del sistema (GRS) adaptado para procesar otro conjunto de frames y usar otro conjunto de plantillas. En consecuencia, en esta sección, se describe únicamente el formato de la información de entrada al módulo, es decir, los frames del usuario, generados por el GDU.

Los frames de respuesta del usuario se construyen, en el módulo simulador, GDU, a partir del estado actual del modelo de diálogo, completando la información de valores de atributos mediante la consulta del registro histórico. En la Tabla 7.1 se lista los identificadores de frame del usuario y su correspondencia con los actos de diálogo (con el 1º nivel del modelo de diálogo):

<i>Identificador de frame</i>	<i>1º nivel etiquetado acto de diálogo</i>
(APERTURA)	Apertura
(CIERRE)	Cierre
(CONFIRMACIÓN)	Confirmación
(AFIRMACIÓN)	Afirmación
(NEGACIÓN)	Negación
(INFORMACIÓN)	Respuesta
(NO-ENTENDIDO)	No-Entendido
(INDEFINIDA)	Indefinida
(SOLICITUD)	Pregunta

Tabla 7.1. Frames del usuario

Todos los frames del usuario van seguidos de su valor de confianza, que es siempre 1.00, porque este valor, actualmente, no se usa. Los frames (CONFIRMACIÓN), (INFORMACIÓN) y (SOLICITUD) irán seguidos por los pares nombre y valor de atributo que corresponda según el estado alcanzado en el modelo de diálogo.

Además, y para permitir que el generador de respuestas pueda realizar aportes espontáneos de datos, los frames del usuario pueden ir seguidos, también, por los pares nombre y valor de atributo de tipo ACTUAL que estén presentes en el registro histórico del gestor de diálogo y hayan sido modificados en el último turno (atributos cuya edad sea menor o igual a 2).

Los atributos usados en los frames del usuario son sólo los siguientes: ORIGEN, DESTINO, TIPO-TREN, PRECIO, TIEMPO-RECORRIDO, SERVICIO, NÚMERO-TRENES, FECHA-SALIDA, HORA-SALIDA, HORA-LLEGADA. Los nombres de atributo para el viaje de vuelta son los mismos seguidos del sufijo -V. Los nombres de atributo de tipo ACTUAL son los mismos seguidos del sufijo -ACTUAL.

A continuación, se ilustra el formato de los frames más importantes, los frames (INFORMACIÓN) y (SOLICITUD), mediante sendos ejemplos:

- Si el estado alcanzado es (U:PREGUNTA:PRECIO), entonces el frame del usuario puede ser:

(SOLICITUD) 1.00

PRECIO: ??? *d.dd*

DESTINO-ACTUAL: *ccc d.dd*

INTERVALO-HORA-SALIDA-ACTUAL: *dd.dd-dd.dd d.dd*

Suponiendo que los valores actuales de ciudad de destino e intervalo de hora de salida se hubieran modificado en el último turno (o que el GDU les hubiera asignado una edad “baja”, menor o igual a 2, con el objeto de que se incluyan como un aporte espontáneo de datos, en una frase como, por ejemplo: “*A Zaragoza, por la mañana, ¿cuánto cuesta?*”).

- Si el estado alcanzado es (U:RESPUESTA:DESTINO, ORIGEN), entonces el frame del sistema puede ser:

(INFORMACIÓN) 1.00

DESTINO: *ccc d.dd*

ORIGEN: *ccc d.dd*

FECHA-ACTUAL: *dd-dd-dddd d.dd*

Suponiendo que la fecha de salida fuera incluida como un aporte espontáneo (por ejemplo, en una frase como la siguiente: “*el día diez de Junio, quiero ir de Valencia a Soria*”).

Como referencia, las Tablas 10.11 y 10.12 del Anexo listan todos los estados del usuario en el actual modelo de diálogo, y su correspondiente secuencia de frames.

8. Evaluación del sistema de diálogo

8. Evaluación del sistema de diálogo

8.1 Introducción

En el apartado 2.5 se presentaron las técnicas desarrolladas por diferentes grupos de investigación para la evaluación de los sistemas de diálogo. Aunque existe algún marco formal para la evaluación de los sistemas, como el paradigma PARADISE, su adaptación a la evaluación de un sistema concreto no resulta sencilla. En consecuencia, en la presente tesis, se ha optado por realizar una evaluación considerando las medidas cuantitativas de uso más extendido: tasas de éxito de los diálogos, duración de los diálogos (número de turnos), tipo de turnos y sus porcentajes, tasas de corrección de errores, etc.

En los apartados 8.2, 8.3 y 8.4 se informa de la evaluación del módulo gestor de diálogo, en funcionamiento con usuarios reales. En esta evaluación, de la que se informa en (Torres et al., 2005a), se investigó el comportamiento del gestor de diálogo ante entradas que contenían errores en diferentes proporciones. Se comprobó que el gestor de diálogo funciona adecuadamente gracias a la incorporación de estrategias de diálogo que tienen en cuenta las medidas de confianza facilitadas por los módulos de comprensión y reconocimiento del habla. En dicha evaluación se usaron los modelos estocásticos de diálogo estimados a partir del corpus de BASURDE.

En el apartado 8.5 se exponen los resultados del reaprendizaje de los modelos iniciales de diálogo (los modelos estimados a partir del corpus de BASURDE) mediante la aplicación de la técnica de simulación del usuario. A partir del modelo inicial, **MD**, se han obtenido tres modelos: **MD*** (resultado del aprendizaje mediante simulaciones con ausencia de errores), **MD*^e** (resultado del aprendizaje mediante simulaciones con presencia de errores y uso del módulo de comprensión expuesto en la Sección 4) y **MD*^j** (resultado del aprendizaje mediante simulaciones con presencia de errores y uso del otro módulo de comprensión). Los cuatro modelos han sido objeto de una evaluación comparativa, obteniéndose unos resultados de mejora razonable, que son objeto de comentario en este mismo apartado.

En el apartado 8.6, se evalúa la importancia de las técnicas de generalización semántica y de las reglas de consistencia con el registro histórico (mediante poda de transiciones incoherentes), ya expuestas en la Sección 5. Se pone de manifiesto que estas técnicas son decisivas en el logro de las buenas prestaciones del gestor de diálogo y que, por tanto, complementan adecuadamente a los modelos estocásticos de diálogo. Por último, en el apartado 8.7, se presenta un breve apunte sobre evaluación comparativa de los modelos **MD** y **MD*^e** en diálogos con usuario real. Los resultados no pueden considerarse concluyentes dado el reducido número de diálogos, 84, ejecutados, y que éstos han sido llevados a cabo por un mismo usuario. Una evaluación fiable de la posible mejora de los modelos **MD***, **MD*^e** o **MD*^j** requerirá una evaluación mucho más amplia en número de diálogos y de usuarios.

8.2 Evaluación del gestor de diálogo

En esta sección, presentamos los resultados de la evaluación del sistema de diálogo. Hemos usado modelos estocásticos, aprendidos del corpus de BASURDE (227 diálogos, 3.528 turnos, 14.902 palabras), en los módulos de comprensión y gestor de diálogo.

Se han evaluado los efectos de facilitar medidas de confianza al gestor de diálogo considerando dos estrategias:

- ❑ Estrategia (a). Consistente en ignorar las medidas de confianza y aceptar como 100% fiable toda la información presente en los frames de usuario. Esta estrategia implica la poda de todas las transiciones a estados de confirmación de cualquier atributo que haya recibido algún valor.
- ❑ Estrategia (b). Consistente en usar las medidas de confianza para podar transiciones o para disminuir o aumentar las probabilidades de las transiciones, tal como se ha explicado.

La estrategia (a) nos lleva, claramente, a diálogos más breves y, quizás, más fluidos, pero implica un importante riesgo de malentendido dado que se prescinde de informar al usuario sobre lo que se ha comprendido de sus intervenciones. La estrategia (b) es más flexible, permitiendo diálogos cortos para obtener la información solicitada cuando las entradas al gestor de diálogo tengan confianzas altas, u obligando a turnos adicionales cuando algunas entradas sean poco fiables.

En los diálogos donde los valores de los atributos se identifiquen correctamente pero con baja fiabilidad, el uso de las medidas de confianza dará lugar a una conversación más larga y tediosa, al obligar al usuario a responder a más confirmaciones explícitas. Éste, sin embargo, es el precio para que no se le den al usuario respuestas equivocadas. Aunque las medidas de confianza no proporcionan una certeza completa acerca de cuáles valores de los atributos son correctos y cuáles no lo son, podemos considerar que una medida de confianza baja es un indicador de un posible error en los procesos de reconocimiento del habla y de comprensión del lenguaje.

A continuación, primero, se presentan dos ejemplos que ilustran las ventajas de la estrategia (b) sobre la estrategia (a), y después, se presentan los resultados experimentales obtenidos al ejecutar 150 diálogos.

8.3 Ejemplos del comportamiento del sistema

En los ejemplos, los turnos Ux indican las frases originales del usuario y los frames de entrada al gestor de diálogo, los turnos Vx son las frases generadas tras la fase de reconocimiento del habla, y los turnos Sx son los frames de salida del gestor de diálogo y las frases de salida del generador de respuestas.

En los dos ejemplos, el escenario es el mismo: consulta de horarios de trenes a Alicante en fin de semana, suponiendo implícitamente que la ciudad de origen es Valencia. En el ejemplo de la estrategia (a), ocurre un error en el turno U1 que causa que el sistema, erróneamente, entienda Sevilla como ciudad de origen y responda informando de horarios de Sevilla a Alicante. El usuario no tiene posibilidad de detectar el malentendido dado que el sistema no realiza ninguna confirmación implícita y, en consecuencia, recibe información falsa. De este modo, el diálogo resulta fallido.

En el ejemplo de la estrategia (b), se presentan dos errores. El primer error aparece en el turno U0, donde el sistema entiende que Málaga es la ciudad origen. Sin embargo, como su medida de confianza es inferior al umbral 0.8, este valor erróneo forma parte de la confirmación implícita integrada en el turno S0. Esto da al usuario la oportunidad de corregir el error en el turno U1. El segundo error ocurre en el mismo turno U1. Esta vez, el sistema interpreta que la ciudad destino es Madrid. De nuevo, mediante la confirmación implícita que forma parte del turno S1, el usuario tiene la posibilidad de detectar el error. Aunque el sistema realiza una consulta innecesaria (por incorrecta) a la base de datos, el usuario puede repetir su petición, obteniendo la información deseada un turno después. El diálogo es un poco más largo, pero a su fin el sistema proporciona la información correcta.

Ejemplo estrategia (a)

U0: *Quería saber los horarios a Alicante en fin de semana.*

V0: Quería los de horarios a Alicante en la.

U0: (HORA-SALIDA) * [1.00]
DESTINO: Alicante * [1.00]

S0: (PREGUNTA)
HORA-SALIDA: ???
DESTINO-ACTUAL: Alicante * [1.00]

S0: *¿Quiere saber la hora de salida?*

U1: *Sí, a Alicante el viernes que viene.*

V1: Cuál sí Sevilla Alicante el viernes que viene.

U1: (AFIRMACIÓN) * [1.00]
() * [1.00]
ORIGEN: Sevilla * [1.00]
DESTINO: Alicante * [1.00]
FECHA-SALIDA: 21-01-2005 * [1.00]

S1: (INFORMACIÓN)
NÚMERO-TRENES: 4
HORA-SALIDA: 13.50
ORIGEN-ACTUAL: Sevilla * [1.00]
(NUEVA-CONSULTA)

S1: *Hay cuatro trenes y el primero sale a las dos menos diez de la tarde. ¿Desea algo más?*

U2: *No, gracias.*

V2: No, gracias.

U2: (CIERRE) * [1.00]

S2: (CIERRE)

S2: *Gracias, buen viaje.*

Ejemplo estrategia (b)

U0: *Quería saber los horarios a Alicante en fin de semana.*

V0: Quería el los horarios Málaga Alicante en fin de semana.

U0: (HORA-SALIDA) * [0.94, 1.00]
 ORIGEN: Málaga * [0.58, 0.60]
 DESTINO: Alicante * [0.73, 0.72]
 FECHA-SALIDA: 21-01-2005 * [0.52, 0.49]

S0: (CONFIRMACIÓN)
 FECHA-SALIDA: 21-01-2005 * [0.49]
 ORIGEN-ACTUAL: Málaga * [0.60]
 DESTINO-ACTUAL: Alicante * [0.72]

S0: *De Málaga a Alicante, ¿quiere viajar el día veintiuno de enero?*

U1: *Sí el veintiuno de enero pero desde Valencia.*

V1: Sí el veintiuno Madrid enero pero desde Valencia.

U1: (AFIRMACIÓN) * [0.97, 1.00]
 () * [0.92, 1.00]
 ORIGEN: Valencia * [0.81, 0.83]
 DESTINO: Madrid * [0.61, 0.63]
 FECHA-SALIDA: 21-01-2005 * [0.56, 0.57]

S1: (CONFIRMACIÓN)
 HORA-SALIDA:???
 ORIGEN-ACTUAL: Valencia * [0.83]
 DESTINO-ACTUAL: Madrid * [0.63]

(ESPERA)

(INFORMACIÓN)

NÚMERO-TRENES: 5

HORA-SALIDA: 14.55

(NUEVA-CONSULTA)

S1: *A Madrid, le consulto la hora de salida. Un momento por favor. Hay 5 trenes y el primero sale a las tres menos cinco de la tarde. ¿Desea algo más?*

U2: *A Madrid no, quería ir a Alicante.*

V2: A Madrid no, quería a Alicante.

U2: () * [0.92, 1.00]
 DESTINO: Alicante * [0.85, 0.87]

S2: (CONFIRMACIÓN)
 HORA-SALIDA: ???
 DESTINO-ACTUAL: Alicante * [0.87]

(ESPERA)

(INFORMACIÓN)

NÚMERO-TRENES: 2

HORA-SALIDA: 02.05

(NUEVA-CONSULTA)

S2: *Le consulto la hora de salida. Un momento por favor. Hay 2 trenes y el primero sale a las dos y cinco de la noche. ¿Desea algo más?*

U3: *No, gracias.* = **V3:** No, gracias.

U3: (CIERRE) * [1.00]

S3: (CIERRE)

S3: *Gracias, buen viaje.*

8.4 Evaluación experimental

Hemos evaluado el comportamiento del gestor de diálogo usando medidas de confianza en 10 escenarios consistentes en consultas sobre horarios y/o precios de viajes de ida. Usuarios expertos adquirieron 150 diálogos (15 diálogos de cada escenario). Se han considerado las siguientes medidas:

- ❑ Tasa de éxito de diálogo (%éxito). Es el porcentaje de diálogos completados satisfactoriamente. En cada escenario, el usuario ha de obtener uno o varios elementos de información. El éxito del diálogo depende de si el sistema proporciona datos correctos (desde el punto de vista de los objetivos del escenario) o datos incorrectos al usuario.
- ❑ Número medio de turnos (nT). Es el número medio de turnos del sistema por diálogo.
- ❑ Tasa de confirmaciones (%confirm). Se obtiene contando el número de turnos de confirmación explícita, nCT, por número de turnos de sistema, nT. Es decir, nCT / nT .
- ❑ Número medio de errores corregidos por diálogo (nCE). Es el número medio de errores detectados y corregidos por el gestor de diálogo. Se han contado sólo los errores que modifican los valores de los atributos, dado que éstos son los que suelen causar el fallo del diálogo.
- ❑ Número medio de errores no corregidos por diálogo (nNCE). Es el número medio de errores no corregidos por el gestor de diálogo. Como en el caso anterior, sólo se han considerado los errores que modifican los valores de los atributos.
- ❑ Tasa de corrección de errores (%correct). Es el porcentaje de errores corregidos. Es decir, $nCE / (nCE + nNCE)$.

Para estudiar el comportamiento del sistema en función del rendimiento del reconocedor del habla, hemos realizado experimentos simulando un reconocedor con diferentes valores de precisión (*word accuracy*, WA). Concretamente, hemos realizado dos series de experimentos: en la primera serie se ha simulado una WA de 80%, en la segunda serie una WA de 70%. Los errores se generaron introduciendo aleatoriamente, pero en las proporciones indicadas (véase la Tabla 8.1), borrados, inserciones y sustituciones en las frases del usuario.

WA	Errores por borrado	Errores por inserción	Errores por sustitución
80,7%	7,5%	4,4%	10,3%
70,7%	12,7%	7,0%	17,0%

Tabla 8.1. Tipo y cantidad de errores introducidos en las series

En nuestros experimentos, hemos verificado el comportamiento del sistema usando la estrategia (b), es decir, con medidas de confianza, y usando la estrategia (a), esto es, sin medidas de confianza. En el caso de la estrategia (b), hemos probado el valor umbral de confianza mínima (UCM) para aceptar los valores de los atributos. Así, el valor de UCM determinará la poda de las transiciones a estados de confirmación de los atributos.

La primera serie de experimentos (con WA de 80%) constó de 90 diálogos. Se realizaron 30 diálogos usando la estrategia (b) y un UCM de 70%; 30 diálogos usando la estrategia (b) y un UCM de 60%; y 30 diálogos usando la estrategia (a). Los resultados se muestran en la Tabla 8.2.

	%éxito	nT	%confirm	nCE	nNCE	%correct
Estrategia (b), UCM = 70%	92	5,63	51	1,06	0,23	82
Estrategia (b), UCM = 60%	99	4,73	37	0,90	0,07	93
Estrategia (a)	63	4,56	37	0,57	0,56	50

Tabla 8.2. Evaluación con WA de 80%

Se puede observar que, cuando el gestor de diálogo usa las medidas de confianza, la tasa de éxito es próxima al 100%. Ignorando las medidas de confianza, en la estrategia (a), la tasa de éxito desciende al 63%. Los diálogos son más largos usando medidas de confianza (4,73 turnos vs. 4,56 turnos) y también usando un UCM superior (5,63 turnos vs. 4,56 turnos). Las tasas de corrección de errores usando medidas de confianza son claramente mejores que no usándolas. A partir de estos resultados, se concluye que un UCM de 60% es adecuado para escenarios del tipo considerado y para este nivel de errores de reconocimiento (WA de 80%). De hecho, en esta serie, la tasa de éxito resultó peor usando un UCM superior (92% vs. 99%). Esto, sin embargo, puede explicarse por el número más alto de errores (1,29 errores vs. 0,97 errores por diálogo), debido a la mayor duración de los diálogos al incrementar el umbral. Por tanto, se decidió usar un UCM de 60% en la siguiente serie de experimentos.

La segunda serie de experimentos (con WA de 70%) constó de 60 diálogos. Se realizaron 30 diálogos usando la estrategia (b) y un UCM de 60%; y 30 diálogos usando la estrategia (a). Los resultados se muestran en la Tabla 8.3.

	%éxito	nT	%confirm	nCE	nNCE	%correct
Estrategia (b), UCM = 60%	69	6,13	47	2,13	0,56	79
Estrategia (a)	45	4,99	33	0,83	1,16	42

Tabla 8.3. Evaluación con WA de 70%

Las ventajas de usar medidas de confianza también resultan apreciables en esta serie. La tasa de éxito es 69% usando medidas de confianza y 45% no usándolas. La tasa de corrección de errores es 79% usando medidas de confianza y 42% no usándolas. Los diálogos son, tan sólo, un poco más largos usando medidas de confianza (6,13 turnos vs. 4,99 turnos).

Finalmente, adquirimos 30 diálogos usando un reconocedor de habla y, por tanto, suministrando al sistema la entrada hablada. 73% es la WA del reconocedor que se usó. Los resultados aparecen en la Tabla 8.4. Se aprecia que los resultados obtenidos anteriormente, usando entrada de texto y un simulador de errores, son coherentes con estos últimos resultados en los que se usó entrada hablada.

	%éxito	nT	%confirm	nCE	nNCE	%correct
Estrategia (b)	84	6,13	49	0,97	0,23	81

Tabla 8.4. Evaluación con entrada hablada

8.5 Evaluación / entrenamiento de los modelos (con simulación de usuario)

Se ha llevado a cabo el aprendizaje de modelos de diálogo estocásticos, usando la técnica de simulación de diálogos descrita. Los resultados muestran la viabilidad de esta técnica.

Se han definido 14 escenarios, consistentes en consultas acerca de horarios, precios y/o tipos de trenes para viajes de ida. En la Tabla 10.3 del Anexo se detallan los 14 escenarios considerados.

En una primera serie de experimentos, se ejecutaron 5.000 simulaciones (aproximadamente, 357 diálogos para cada escenario). El modelo de diálogo inicial, MD, usado tanto por el gestor de diálogo (en su versión dinámica o adaptativa, dMD) y por el simulador de usuario (en su versión estática, eMD), fue el modelo estocástico inferido a partir de los 227 diálogos reales del corpus de BASURDE. En esta serie, la entrada al gestor de diálogo (los frames generados por el módulo de comprensión) era la representación semántica exacta, es decir, sin presencia de errores, de las frases generadas en el GRU.

Sea MD*, el modelo aprendido, partiendo del modelo inicial de BASURDE y después de ejecutar estas 5.000 simulaciones. En la Tabla 8.5 se resumen algunos datos estadísticos relevantes de las simulaciones de aprendizaje del modelo MD*.

Aprendizaje del modelo MD*	Nº total	%	Nº medio por diálogo
Diálogos	5.000		
Diálogos concluidos con éxito	4.931	98,62%	
Turnos	23.626		4,73
Turnos (sistema) de Respuesta	11.897		2,38
Turnos (sistema) de Confirmación y Pregunta	357		0,07
Turnos (sistema) de Confirmación	7.713		1,54
Turnos (sistema) de Pregunta	2.888		0,58
Turnos (sistema) de Confirmación y Espera	368		0,07
Turnos (sistema) de sólo Nueva-Consulta	142		0,03
Turnos (sistema) de sólo No-Entendido	48		0,01
Transiciones ordinarias a estados usuario	23.574	99,78%	
Transiciones extraordinarias a estados usuario	52	0,22%	
Transiciones ordinarias a estados sistema	23.413	99,10%	
Transiciones extraordinarias a estados sistema	213	0,90%	

Tabla 8.5: Aprendizaje del modelo MD*

En la figura 8.1 se muestra la evolución de las probabilidades de algunas transiciones durante esta serie de experimentos. Se han elegido porque ilustran algunos cambios interesantes:

- Hay transiciones razonables con probabilidades significativas en el modelo nuevo, MD*, y que no existían (es decir, tenían probabilidad cero) en el modelo inicial,

MD. Por ejemplo, las transiciones desde un estado de no-entendido y petición de confirmación del atributo destino o del atributo origen, por parte del sistema, a un estado de aceptación del atributo destino o del atributo origen, por parte del usuario.

- Se encuentran también otras transiciones razonables que aparecen en ambos modelos, pero con una probabilidad más alta en el MD*. Por ejemplo, la transición desde un estado de respuesta de los horarios de salida, por parte del sistema, a un estado de pregunta de los horarios de llegada, por parte del usuario.
- Por otra parte, hay, lógicamente, otras transiciones con menos probabilidad en el MD* que en el MD. Por ejemplo, a la petición de confirmación de los atributos origen y destino, por parte del sistema, el simulador de usuario no suele contestar

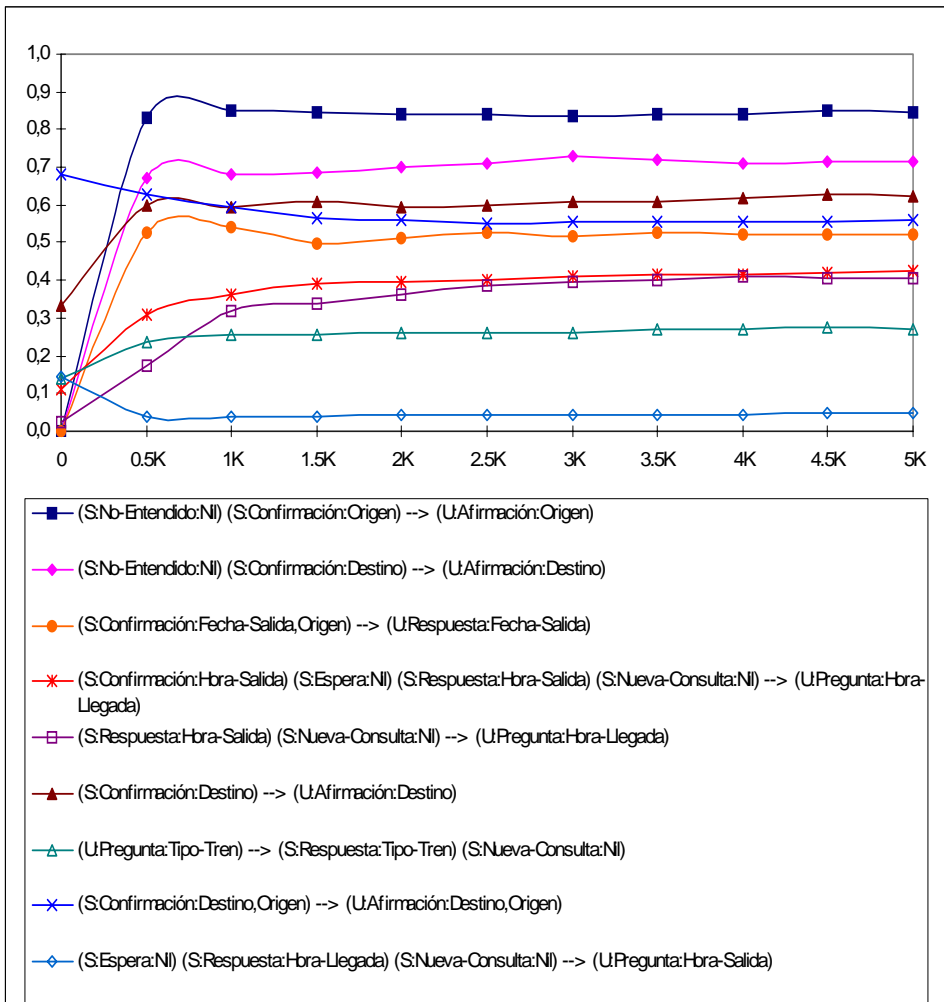


Figura 8.1: Evolución de las probabilidades de algunas transiciones durante el entrenamiento mediante simulación de diálogos (aprendizaje del modelo MD*)

aceptando los dos valores sugeridos (es decir, habitualmente uno de los dos valores debe ser corregido). Otro ejemplo es la disminución de probabilidad de que el usuario pida horarios de salida después de conocer los horarios de llegada, caso que se explica por la sencillez de los escenarios considerados (donde sólo se buscan los horarios de un tren; si se simularan escenarios más complejos, donde se pidieran horarios de trenes anteriores o posteriores al facilitado por el sistema, la evolución de la probabilidad de esta transición podría ser diferente).

En la Figura 8.1 se puede apreciar, además, que las probabilidades tienden a converger después de ejecutarse 1.000 simulaciones aproximadamente. Por tanto, se puede concluir que la técnica propuesta permite una adaptación del modelo de diálogo, dinámica y estable, puesto que converge tras un número moderado de iteraciones.

Con la intención de comparar el comportamiento del sistema usando MD o MD* en el gestor de diálogo, hemos llevado a cabo 1.000 nuevas simulaciones con la única variación de usar uno u otro modelo. En la Tabla 8.6 se muestra los resultados más relevantes de la comparativa.

La tasa de éxito (porcentaje de diálogos en los que el sistema reconoce correctamente los datos facilitados por el simulador y, por tanto, le proporciona la información solicitada) es próxima al 100% dado que no se introducían errores. Se aprecia que, usando MD*, los diálogos son algo más breves, que nunca se transita a estados inadecuados para una interacción con usuarios reales como los de ESPERA y NO-ENTENDIDO, y que apenas se usa el modelo auxiliar, MD-aux (es decir, MD* casi siempre dispone de transiciones válidas).

Para observar el comportamiento del sistema usando MD o MD* en situaciones con alta tasa de errores, hemos realizado otras 500 simulaciones introduciendo errores en la identificación de los atributos ORIGEN, DESTINO y FECHA-SALIDA, cuya corrección es determinante del éxito de los diálogos. En la Tabla 8.7 se muestra los resultados de esta nueva comparativa.

	MD	MD*
Nº de diálogos	1.000	1.000
Tasa de éxito	98,90%	98,90%
Nº medio errores por diálogo	0	0
Nº medio turnos por diálogo	4,75	4,61
Nº total de turnos	4.748	4.613
Nº total de turnos a estados de RESPUESTA	2.385	2.380
Nº total de turnos a estados de CONFIRMACIÓN y/o PREGUNTA	2.235	2.222
Nº total de turnos a estados de ESPERA y de NO-ENTENDIDO	49	0
Nº total de turnos a estados de NUEVA-CONSULTA	20	11
Nº total de turnos a otros estados	59	0
Nº total de llamadas a MD-aux	68	11

Tabla 8.6: Primera comparativa, MD y MD*

	MD	MD*
Nº de diálogos	500	500
Tasa de éxito	75,40%	76,40%
Nº medio errores por diálogo	0,86	0,79
Nº medio turnos por diálogo	4,84	4,84
Nº total de turnos	2.422	2.420
Nº total de turnos a estados de RESPUESTA	1.197	1.182
Nº total de turnos a estados de CONFIRMACIÓN y/o PREGUNTA	1.146	1.229
Nº total de turnos a estados de ESPERA y de NO-ENTENDIDO	31	3
Nº total de turnos a estados de NUEVA-CONSULTA	30	6
Nº total de turnos a otros estados	18	0
Nº total de llamadas a MD-aux	24	5

Tabla 8.7: Segunda comparativa, MD y MD*

Se observa que la introducción de, casi, un error grave por diálogo supone sólo un leve incremento en la duración media y que la tasa de éxito desciende a valores en torno al 75%. Los resultados de esta segunda experimentación indican que MD responde algo mejor ante los errores que MD* (las tasas de éxito difieren en un 1%, pero en las simulaciones con MD se generaron más errores). Esto se explicaría por el hecho de que no se introdujeron errores en las 5.000 simulaciones con las que, a partir de MD, se generó MD*. En consecuencia, la estrategia aprendida en MD* resulta demasiado optimista en situaciones de alta tasa de errores.

En la Figura 8.2 se muestra un ejemplo de diálogo de la segunda experimentación usando MD*. En los turnos de usuario, numerados con Ux , se indica los frames generados por el GDU, las frases generadas por el GRU, y los frames generados por el módulo de comprensión donde se introducían los errores. En los turnos de sistema, numerados con Sx , se indica los frames generados por el GDS y las frases generadas por el GRS. En la comprensión del turno U1 se introdujo un error, cambiando la ciudad origen *Toledo* por *Oviedo*. Este error pudo corregirse mediante el turno S1, de confirmación del sistema, y el turno U2, de negación y corrección del simulador de usuario. En la comprensión del turno U3 se introdujo otro error, cambiando la fecha de salida *cuatro de Noviembre* por *cuatro de Diciembre*. Este error podría haber sido detectado mediante la confirmación implícita incluida en el inmediato turno S3. Sin embargo, la corrección se retrasó hasta la respuesta del usuario, en el turno U5, a la pregunta explícita del sistema acerca del valor del atributo fecha de salida (turno S4).

U0: (HORA-SALIDA) 1.00
 FECHA-ACTUAL: 04-11-???? 1.00
¿A qué hora sale?
 (HORA-SALIDA) 0.36

S0: (NO-ENTENDIDO) 1.00
 (CONFIRMACIÓN) 1.00
 ORIGEN: Valencia 0.62
 DESTINO: Zaragoza 0.63
Disculpe pero no le he entendido. ¿Quiere ir de Valencia a Zaragoza?

U1: (NEGACIÓN) 1.00
 () 1.00
 ORIGEN: Toledo 0.62
 DESTINO: Salamanca 0.38
No. Quiero ir de Toledo a Salamanca.
 (HORA-SALIDA) 0.36
 ORIGEN: Oviedo 0.39
 DESTINO: Salamanca 0.77

S1: (CONFIRMACIÓN) 1.00
 ORIGEN: Oviedo 0.56
 DESTINO-ACTUAL: Salamanca 0.77
A Salamanca. ¿Quiere salir de Oviedo?

U2: (NEGACIÓN) 1.00
 () 1.00
 ORIGEN: Toledo 0.62
 DESTINO: Salamanca 0.38
No. Quiero ir de Toledo a Salamanca.
 (HORA-SALIDA) 0.36
 ORIGEN: Toledo 0.35
 DESTINO: Salamanca 0.31

S2: (CONFIRMACIÓN) 1.00
 ORIGEN: Toledo 0.40
 FECHA-SALIDA: 22-04-2005 0.12
¿Quiere viajar desde Toledo el veintidós de Abril?

U3: (NEGACIÓN) 1.00
 () 1.00
 FECHA-SALIDA: 04-11-???? 0.68
No. Quiero viajar el cuatro de Noviembre.
 (NEGACIÓN) 0.58
 () 0.00
 FECHA-SALIDA: 04-12-???? 0.21

S3: (SOLICITUD) 1.00
 ORIGEN: Toledo 0.71

FECHA-ACTUAL: 04-12-???? 0.21
El día cuatro de Diciembre. ¿De dónde quiere salir?

U4: (INFORMACIÓN) 1.00
 ORIGEN: Toledo 0.62
 DESTINO: Salamanca 0.38
Quiero ir de Toledo a Salamanca.
 (HORA-SALIDA) 0.43
 ORIGEN: Toledo 0.23
 DESTINO: Salamanca 0.25

S4: (SOLICITUD) 1.00
 FECHA-SALIDA: 04-12-???? 0.58
¿Qué día quiere salir?

U5: () 1.00
 FECHA-SALIDA: 04-11-???? 0.68
Quiero viajar el cuatro de Noviembre.
 () 0.00
 FECHA-SALIDA: 04-11-???? 0.63

S5: (CONFIRMACIÓN) 1.00
 HORA-SALIDA: ??? 0.00
 FECHA-ACTUAL: 04-11-???? 0.63
 (ESPERA) 1.00
 (INFORMACIÓN) 1.00
 NÚMERO-TRENES: 06 0.99
 HORA-SALIDA: 08.14 0.99
 FECHA-ACTUAL: 04-11-???? 0.63
 (NUEVA-CONSULTA) 1.00
El día cuatro de Noviembre. Hay seis trenes, y le informo del primero. Le consulto la hora de salida. Un momento por favor. El tren sale a las ocho y catorce de la mañana. ¿Desea algo más?

U6: (HORA-LLEGADA) 1.00
¿A qué hora llega?
 (HORA-LLEGADA) 0.52

S6: (ESPERA) 1.00
 (INFORMACIÓN) 1.00
 HORA-LLEGADA: 12.14 0.99
 (NUEVA-CONSULTA) 1.00
Un momento por favor. El tren llega a las doce y catorce de la mañana. ¿Desea algo más?

U7: (CIERRE) 1.00
Nada más, muchas gracias

Figura 8.2: Ejemplo de diálogo

Como se ha apuntado, el modelo inicial, MD, parece responder algo mejor ante los errores que MD*. Este hecho podría deberse a que la estrategia aprendida en MD*, tras 5.000 simulaciones libres de error, es demasiado optimista y no resulta adecuada para afrontar situaciones con alta tasa de errores. Por ello, se decidió repetir el proceso de aprendizaje introduciendo errores en los atributos ORIGEN, DESTINO y FECHA-SALIDA, aleatoriamente, pero en la misma proporción en que después, se volverían a evaluar los modelos.

Sea MD*^c, el modelo aprendido, partiendo del modelo inicial de BASURDE y después de ejecutar estas 5.000 simulaciones con introducción de errores. En la Tabla 8.8 se resumen algunos datos estadísticos relevantes de las simulaciones de este aprendizaje. De las 3.745 simulaciones concluidas con éxito (es decir, con igualdad de contenidos de RHU y RHS y consiguiente consolidación de las transiciones realizadas en el modelo MD*^c), 2.465 corresponden a diálogos donde no se introdujeron errores y 1.280 a diálogos donde los errores introducidos fueron detectados y corregidos.

Aprendizaje del modelo MD* ^c	Nº total	%	Nº medio por diálogo
Diálogos	5.000		
Diálogos sin introducción de errores	2.465	49,30%	
Diálogos con introducción de errores	2.535	50,70%	
Diálogos concluidos con éxito	3.745	74,90%	
Errores en atributo ORIGEN	1.289		0,26
Errores en atributo DESTINO	1.644		0,33
Errores en atributo FECHA-SALIDA	1.281		0,26
Turnos	24.513		4,90
Turnos (sistema) de Respuesta	11.937		2,39
Turnos (sistema) de Confirmación y Pregunta	406		0,08
Turnos (sistema) de Confirmación	8.111		1,62
Turnos (sistema) de Pregunta	3.331		0,67
Turnos (sistema) de Confirmación y Espera	338		0,07
Turnos (sistema) de sólo Nueva-Consulta	152		0,03
Turnos (sistema) de sólo No-Entendido	37		0,01
Transiciones ordinarias a estados usuario	24.440	99,70%	
Transiciones extraordinarias a estados usuario	73	0,30%	
Transiciones ordinarias a estados sistema	24.312	99,18%	
Transiciones extraordinarias a estados sistema	201	0,82%	

Tabla 8.8: Aprendizaje del modelo MD*^c

En la Figura 8.3 se muestra la evolución de las probabilidades de las mismas transiciones consideradas en la anterior Figura 8.1. Al igual que sucedió con el modelo MD*, en el aprendizaje del modelo MD*^c se aprecia que la convergencia de las probabilidades a partir, aproximadamente, de 1.000 iteraciones.

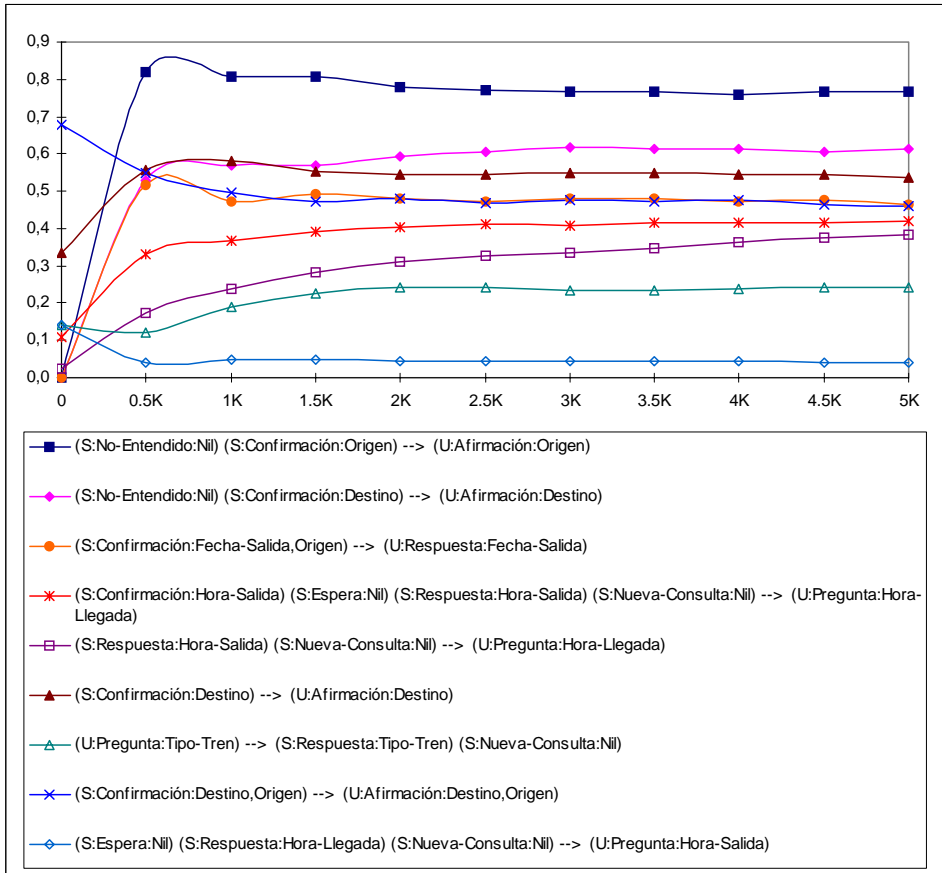


Figura 8.3: Evolución de las probabilidades de algunas transiciones durante el entrenamiento mediante simulación con introducción de errores (aprendizaje del modelo MD^{*c})

En las gráficas de la Figura 10.1 del Anexo se compara la evolución de las probabilidades de las mismas transiciones en los modelos MD^{*} y MD^{*c}. Se puede apreciar que las transiciones desde estados de confirmación de atributos, por parte del sistema, a estados de afirmación (aceptación) de esos atributos, por parte del usuario, tienen una probabilidad menor en MD^{*c} que en MD^{*}. Este resultado es una consecuencia natural de la introducción de errores en el aprendizaje del modelo MD^{*c} que obliga al simulador de usuario a corregir los atributos erróneos, en lugar de aceptarlos.

De nuevo, para observar el comportamiento del sistema usando los distintos modelos (MD, MD^{*} y MD^{*c}) en situaciones con alta tasa de errores, hemos realizado otras 4 series de 500 simulaciones introduciendo errores en la identificación de los atributos ORIGEN, DESTINO y FECHA-SALIDA. En las Tablas 10.4, 10.5 y 10.6 {a, b, c, d} del Anexo, se muestra los resultados de esta nueva comparativa.

	MD	MD*	MD**^e
Nº de diálogos	2.000	2.000	2.000
Tasa de éxito	75,75%	74,15%	75,70%
Nº medio errores por diálogo	0,84	0,87	0,86
% diálogos con introducción de errores	50,45%	51,05%	50,75%
Tasa de éxito (en diálogos con errores)	51,86%	49,34%	52,15%
Nº medio errores por diálogo (en diálogos con errores)	1,68	1,71	1,68
Tasa de éxito normalizada (en diálogos con errores)	51,86%	50,22%	52,15%
Nº medio turnos por diálogo	4,90	4,78	4,80
Nº total de turnos	9.797	9.562	9.619
Nº total de turnos a estados de RESPUESTA	4.764	4.752	4.760
Nº total de turnos a estados de CONFIRMACIÓN y/o PREGUNTA	4.690	4.765	4.834
Nº total de turnos a estados de ESPERA y de NO-ENTENDIDO	197	28	7
Nº total de turnos a estados de NUEVA-CONSULTA	68	17	17
Nº total de turnos a otros estados	78	0	1
Nº total de llamadas a MD-aux	99	12	9

Tabla 8.9: Tercera comparativa, MD, MD* y MD**^e

En la Tabla 8.9, se resumen los datos más relevantes de las citadas tablas del Anexo. Dado que en el conjunto de esta evaluación (2.000 diálogos) de los modelos se introdujo prácticamente el mismo número de errores, se aprecia cierta superioridad de MD**^e. En tasa de éxito (en diálogos con errores), el MD**^e supera en un 0,29% al MD y, si se normaliza la tasa de diálogos con errores, en un 1,9% al MD*. La duración del diálogo es 0,1 turnos inferior usando MD**^e, lo que se explica, fundamentalmente, por la presencia de un 2,01% de turnos a estados de ESPERA y de NO-ENTENDIDO en MD frente a 0,07% en MD**^e. Además, MD necesita llamar a MD-aux en un 1,01% de turnos frente a sólo un 0,09% de turnos en MD**^e. Estos hechos confirman la hipótesis de que el entrenamiento con introducción de errores permite mejorar el comportamiento del sistema cuando se encuentre en situaciones donde se presenten errores similares en tipo y cantidad a los ocurridos durante el aprendizaje.

Además, se ha observado el comportamiento del sistema usando los modelos MD y MD**^e, sustituyendo el módulo de comprensión expuesto en la Sección 4 por el módulo de comprensión (Segarra et al., 2002). Aunque se han simulado errores en los atributos ORIGEN, DESTINO y FECHA-SALIDA en la misma proporción que en los anteriores experimentos, al utilizar este módulo de comprensión, se ha presentado un porcentaje bastante mayor de situaciones con errores. Se han realizado 2 series de 500 simulaciones. En las Tablas 10.7 y 10.8 {a, b} del Anexo se muestra los resultados de esta nueva comparativa. En la Tabla 8.10 se resume los datos más relevantes de las mismas.

	MD	MD* ^e
Nº de diálogos	1.000	1.000
Tasa de éxito	30,60%	28,10%
Nº medio errores por diálogo	1,63	1,60
% diálogos con introducción de errores	88,10%	89,00%
Tasa de éxito (en diálogos con errores)	21,22%	18,39%
Nº medio errores por diálogo (en diálogos con errores)	1,85	1,80
Tasa de éxito normalizada (en diálogos con errores)	21,22%	17,89%
Nº medio turnos por diálogo	4,09	3,94
Nº total de turnos	4.094	3.946
Nº total de turnos a estados de RESPUESTA	2.353	2.363
Nº total de turnos a estados de CONFIRMACIÓN y/o PREGUNTA	1.544	1.552
Nº total de turnos a estados de ESPERA y de NO-ENTENDIDO	66	6
Nº total de turnos a estados de NUEVA-CONSULTA	42	10
Nº total de turnos a otros estados	89	15
Nº total de llamadas a MD-aux	182	71

Tabla 8.10: Cuarta comparativa, MD y MD*^e

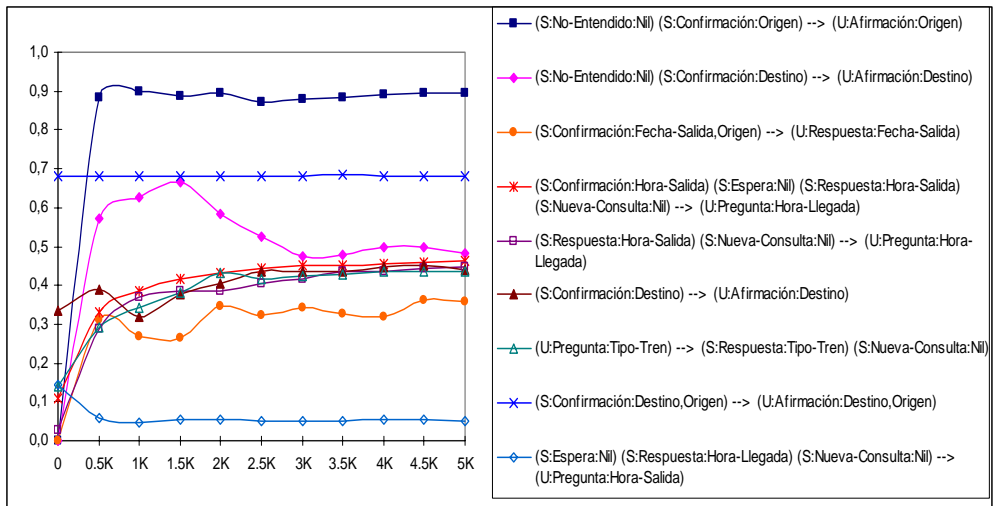
En esta evaluación, si se normaliza la tasa de diálogos con errores, se observa que el MD*^e alcanza una tasa de éxito (en diálogos con errores) 3,33% inferior a la obtenida con MD. La duración del diálogo es 0,15 turnos inferior usando MD*^e, lo que se explica, fundamentalmente, por la presencia de un 1,61% de turnos a estados de ESPERA y de NO-ENTENDIDO en MD frente a 0,15% en MD*^e. Además, MD necesita llamar a MD-aux en un 4,45% de turnos frente a un 1,80% de turnos en MD*^e. Estos hechos apuntan a que las ventajas del entrenamiento con una determinada tasa de introducción de errores no se mantienen si en la evaluación se incrementa apreciablemente la tasa de errores.

Por tanto, se ha procedido al aprendizaje de un nuevo modelo, partiendo del modelo inicial de BASURDE, mediante la ejecución de 5.000 simulaciones con introducción de errores y uso del módulo de comprensión (Segarra et al., 2002).

En la Tabla 8.11 se resumen algunos datos estadísticos relevantes de las simulaciones de este aprendizaje del modelo que llamamos MD*^j. Se introdujeron errores en 4.865 simulaciones y 1.029 simulaciones concluyeron con éxito (es decir, con igualdad de contenidos de RHU y RHS y consiguiente consolidación de las transiciones realizadas en el modelo MD*^j). El número medio de errores introducidos por diálogo fue de 2,08.

En la Figura 8.4 se aprecia, de nuevo, que la evolución de las probabilidades de las transiciones converge después de las 5.000 simulaciones. Hay que destacar que el éxito se ha obtenido en un 20% de simulaciones. Por tanto, como en los otros modelos, con aproximadamente 1.000 simulaciones concluidas con éxito se alcanza la convergencia.

Aprendizaje del modelo MD ^{*j}	Nº total	%	Nº medio por diálogo
Diálogos	5.000		
Diálogos sin introducción de errores	135	2,72%	
Diálogos con introducción de errores	4.865	97,28%	
Diálogos concluidos con éxito	1.029	20,58%	
Errores en atributo ORIGEN	3.639		0,73
Errores en atributo DESTINO	3.671		0,73
Errores en atributo FECHA-SALIDA	3.106		0,62
Turnos	19.192		3,84
Turnos (sistema) de Respuesta	11.841		2,37
Turnos (sistema) de Confirmación y Pregunta	182		0,04
Turnos (sistema) de Confirmación	3.418		0,68
Turnos (sistema) de Pregunta	3.107		0,62
Turnos (sistema) de Confirmación y Espera	240		0,05
Turnos (sistema) de sólo Nueva-Consulta	217		0,04
Turnos (sistema) de sólo No-Entendido	187		0,04
Transiciones ordinarias a estados usuario	18.916	98,56%	
Transiciones extraordinarias a estados usuario	276	1,44%	
Transiciones ordinarias a estados sistema	18.768	97,79%	
Transiciones extraordinarias a estados sistema	424	2,21%	

Tabla 8.11: Aprendizaje del modelo MD^{*j}Figura 8.4: Evolución de las probabilidades de algunas transiciones durante el entrenamiento mediante simulación con introducción de errores (aprendizaje del modelo MD^{*j})

A continuación, se ha observado el comportamiento del sistema usando los modelos MD, MD*^e y MD*^j, usando el módulo de comprensión (Segarra et al., 2002) y simulando errores en los atributos ORIGEN, DESTINO y FECHA-SALIDA. Se han realizado 2.000 simulaciones con cada modelo. En la Tabla 8.12 se resume los datos más relevantes de las mismas.

	MD	MD* ^e	MD* ^j
Nº de diálogos	2.000	2.000	2.000
Tasa de éxito	21,40%	22,10%	22,90%
Nº medio errores por diálogo	2,08	2,04	2,04
% diálogos con introducción de errores	97,50%	96,95%	96,30%
Tasa de éxito (en diálogos con errores)	20,86%	21,43%	22,05%
Nº medio errores por diálogo (en diálogos con errores)	2,03	1,98	1,96
Tasa de éxito normalizada (en diálogos con errores)	20,86%	20,90%	21,29%
Nº medio turnos por diálogo	3,82	3,62	3,66
Nº total de turnos	7.649	7.235	7.326
Nº total de turnos a estados de RESPUESTA	4.724	4.761	4.751
Nº total de turnos a estados de CONFIRMACIÓN y/o PREGUNTA	2.683	2.440	2.457
Nº total de turnos a estados de ESPERA y de NO-ENTENDIDO	150	17	74
Nº total de turnos a estados de NUEVA-CONSULTA	92	17	44
Nº total de llamadas a MD-aux	314	82	100

Tabla 8.12: Comparativa, MD, MD*^e y MD*^j

En esta comparativa se aprecia cierta superioridad de los modelos MD*^e y MD*^j respecto al modelo original, MD. Con MD*^j se obtiene una tasa de éxito (en diálogos con errores) 1,19% superior a la alcanzada con MD, con un promedio de errores prácticamente idéntico en ambas series (en MD es sólo 0,07 superior). Con MD*^e, la tasa de éxito (en diálogos con errores) es 0,57% superior a la alcanzada con MD.

Si se compara con MD, la duración de los diálogos es 0,16 turnos menor usando MD*^j y es 0,2 turnos menor usando MD*^e. Por último, al igual que sucedía con los otros modelos aprendidos mediante simulación, MD*^j transita con mucha menos frecuencia a los estados tipo ESPERA, NO-ENTENDIDO y NUEVA-CONSULTA (los estados menos convenientes para el adecuado progreso de los diálogos) y, además, requiere tres veces menos el concurso del modelo auxiliar, MD-aux.

En todas las anteriores evaluaciones, el simulador de usuario aprovecha las confirmaciones explícitas y las preguntas del sistema para proporcionar los datos pedidos y, en su caso, corregir los errores del sistema. Sin embargo, no atiende a las confirmaciones implícitas del sistema, sino que se limita a aportar espontáneamente otros datos en sus respuestas. El aporte espontáneo de información en el GDU se rige por los criterios expuestos en la Sección 7 y sólo de manera fortuita se produce la corrección de datos erróneos que el GDS haya incluido en sus confirmaciones implícitas.

Se ha considerado ampliar la colaboración del simulador de usuario al tratamiento de las confirmaciones implícitas. Por tanto, se ha modificado el GDU para que atienda a las con-

firmaciones implícitas del GDS. Así, aparte de los atributos que sean objeto de aporte espontáneo, el GDU incluirá en cada nueva intervención los valores de los atributos que el GDS haya citado en el turno precedente con valores erróneos.

A continuación, se ha repetido la evaluación de los modelos MD, MD*^e y MD*^j, usando el módulo de comprensión (Segarra et al., 2002). Se han realizado una serie de 2.000 simulaciones. En la Tabla 8.13 se resumen los resultados obtenidos.

	MD	MD* ^e	MD* ^j
Nº de diálogos	2.000	2.000	2.000
Tasa de éxito	90,80%	92,40%	92,75%
Nº medio errores por diálogo	2,11	2,12	2,09
% diálogos con introducción de errores	97,60%	97,50%	96,80%
Tasa de éxito (en diálogos con errores)	88,62%	90,09%	89,78%
Nº medio errores por diálogo (en diálogos con errores)	2,06	2,07	2,02
Tasa de éxito normalizada (en diálogos con errores)	88,62%	90,53%	88,04%
Nº medio turnos por diálogo	3,93	3,73	3,76
Nº total de turnos	7.851	7.452	7.513
Nº total de turnos a estados de RESPUESTA	4.734	4.739	4.746
Nº total de turnos a estados de CONFIRMACIÓN y/o PREGUNTA	2.857	2.675	2.640
Nº total de turnos a estados de ESPERA y de NO-ENTENDIDO	188	28	56
Nº total de turnos a estados de NUEVA-CONSULTA	72	10	71
Nº total de llamadas a MD-aux	373	169	121

Tabla 8.13: Comparativa, MD, MD*^e y MD*^j

Si se normaliza la tasa de diálogos con errores, el MD*^j alcanza una tasa de éxito (en diálogos con errores) 0,58% inferior a la obtenida con MD. La duración del diálogo es 0,17 turnos inferior usando MD*^j, lo que se explica, fundamentalmente, por la presencia de un 3,31% de turnos a estados de ESPERA, de NO-ENTENDIDO, de NUEVA-CONSULTA, en MD frente a 1,69% en MD*^j. Además, MD necesita llamar a MD-aux en un 4,75% de turnos frente a un 1,61% de turnos en MD*^j. En cuanto al MD*^e, y también si se normaliza la tasa de diálogos con errores, se consigue una tasa de éxito (en diálogos con errores) 1,91% superior a la obtenida con MD. La duración del diálogo es 0,2 turnos inferior usando MD*^e, lo que se explica, fundamentalmente, por la presencia de un 3,31% de turnos a estados de ESPERA, de NO-ENTENDIDO, de NUEVA-CONSULTA, en MD frente a 0,51% en MD*^e. Además, MD necesita llamar a MD-aux en un 4,75% de turnos frente a un 2,27% de turnos en MD*^e.

En consecuencia, al mejorar la estrategia de colaboración del simulador de usuario, se confirma también que el entrenamiento con introducción de errores (en el caso del MD*^e) permite mejorar el comportamiento del sistema.

Por otra parte, hay una evidente mejora en tasa de éxito al incluir el tratamiento de las confirmaciones implícitas del sistema en la estrategia del simulador de usuario. Contrastando

las Tablas 8.12 y 8.13, se aprecia que la tasa de éxito en diálogos con errores, usando MD, se incrementa un 67,76% (de 20,86% a 88,62%), usando MD*^e, se incrementa un 68,66% (21,43% a 90,09%), y usando MD*^j, se incrementa un 67,73% (22,05% a 89,78%).

Sin embargo, este grado de mejora ($\approx 68\%$) al considerar las confirmaciones implícitas no parece que se pueda alcanzar con usuarios humanos. Ahora, la estrategia del simulador de usuario incluye detectar todos los errores que aparezcan en las confirmaciones implícitas del sistema y corregirlos sin excepción. En cambio, los usuarios reales pueden sentirse confundidos ante una petición de confirmación mixta y no responder a la misma con igual eficacia, e incluso es fácil que, por falta de suficiente atención, no detecten cierto porcentaje de errores en las confirmaciones implícitas del sistema. Posiblemente, el comportamiento real de los usuarios se encuentre en un punto medio de las dos estrategias del simulador: la estrategia de aporte espontáneo de información semi-aleatorio (Tabla 8.12) y la estrategia de verificación exhaustiva de las confirmaciones implícitas del sistema (Tabla 8.13).

A continuación, se ha repetido la evaluación de los modelos MD y MD*^e, usando el módulo de comprensión expuesto en la Sección 4. Se ha realizado una serie de 1.000 simulaciones, con la estrategia mejorada de colaboración del usuario (comprobación de las confirmaciones implícitas). En las Tablas 10.9 y 10.10 del Anexo se muestra los resultados de esta nueva comparativa. En la Tabla 8.14 se resumen los resultados obtenidos.

	MD	MD* ^e
Nº de diálogos	1.000	1.000
Tasa de éxito	92,30%	95,40%
Nº medio errores por diálogo	1,17	1,05
% diálogos con introducción de errores	56,40%	51,90%
Tasa de éxito (en diálogos con errores)	86,35%	91,14%
Nº medio errores por diálogo (en diálogos con errores)	2,08	2,02
Tasa de éxito normalizada (en diálogos con errores)	86,35%	88,51%
Nº medio turnos por diálogo	4,54	4,43
Nº total de turnos	4.539	4.427
Nº total de turnos a estados de RESPUESTA	2.400	2.414
Nº total de turnos a estados de CONFIRMACIÓN y/o PREGUNTA	2.006	2.006
Nº total de turnos a estados de ESPERA y de NO-ENTENDIDO	60	2
Nº total de turnos a estados de NUEVA-CONSULTA	37	5
Nº total de turnos a otros estados	36	0
Nº total de llamadas a MD-aux	47	6

Tabla 8.14: Quinta comparativa, MD y MD*^e

Si se normaliza la tasa de diálogos con errores, el MD*^e alcanza una tasa de éxito (en diálogos con errores) 2,16% superior a la obtenida con MD. La duración del diálogo es 0,11 turnos inferior usando MD*^e, lo que se explica, fundamentalmente, por la presencia de un 2,93% de turnos a estados de ESPERA, de NO-ENTENDIDO, de NUEVA-CONSULTA, y otros,

en MD frente a 0,16% en MD*^e. Además, MD necesita llamar a MD-aux en un 1,04% de turnos frente a un 0,16% de turnos en MD*^e.

En consecuencia, al mejorar la estrategia de colaboración del simulador de usuario, se confirma también que el entrenamiento con introducción de errores permite mejorar el comportamiento del sistema.

Por otra parte, hay una evidente mejora en tasa de éxito al incluir el tratamiento de las confirmaciones implícitas del sistema en la estrategia del simulador de usuario. Contrastando las Tablas 8.9 y 8.14, se aprecia que la tasa de éxito en diálogos con errores, usando MD, se incrementa un 34,67% (de 51,86% a 86,35%), y usando MD*^e, se incrementa un 36,36% (52,15% a 88,51%).

8.6 Evaluación de la generalización semántica y la poda de transiciones

En la Sección 5, se describieron dos técnicas para resolver ciertas situaciones que no estaban adecuadamente modelizadas en MD: 1º) la generalización semántica de los frames; 2º) la poda de transiciones de acuerdo a reglas de consistencia con el RH. En este apartado, se han investigado los efectos del uso de estas dos técnicas en el gestor de diálogo.

En Tabla 8.15 se comparan los resultados de sendas simulaciones cuya única diferencia radica en la activación o desactivación de los procedimientos de generalización semántica. En ambos casos, se ha usado el modelo de diálogo MD, el módulo de comprensión (Segarra et al., 2002), con simulación de errores, y la estrategia con verificación de confirmaciones implícitas en el simulador de usuario¹.

	Sin GS	Con GS
Nº de diálogos	300	300
Tasa de éxito	69,33%	91,67%
Nº medio errores por diálogo	2,13	2,11
% diálogos con introducción de errores	96,67%	98,00%
Tasa de éxito (en diálogos con errores)	67,02%	89,84%
Nº medio errores por diálogo (en diálogos con errores)	2,06	2,07
Nº medio turnos por diálogo	5,79	3,90
Nº total de turnos	1.737	1.169
Nº total de turnos a estados de RESPUESTA	918	710
Nº total de turnos a estados de CONFIRMACIÓN y/o PREGUNTA	654	416
Nº total de turnos a estados de ESPERA y de NO-ENTENDIDO	151	31
Nº total de turnos a estados de NUEVA-CONSULTA	14	12
Nº total de llamadas a MD-aux	224	46

Tabla 8.15: Comparativa sobre generalización semántica

Como se aprecia, la contribución de las técnicas de generalización semántica (GS) es decisiva en el desarrollo de los diálogos. En las mismas condiciones experimentales, cuando no se usa GS se obtiene una tasa de éxito 22,82% menor (desciende de 89,84% a 67,02%), una duración media de los diálogos 1,89 turnos mayor (aumenta de 3,90 a 5,79 turnos), y se incrementa la consulta del modelo auxiliar, MD-aux, en un 8,97% (aumenta de 3,93% a 12,90%).

En Tabla 8.16 se comparan los resultados de sendas simulaciones cuya única diferencia radica en la activación o desactivación de los procedimientos de poda de transiciones de acuerdo a reglas de consistencia con el RH. En ambos casos, se ha usado el modelo de

¹ En la Tabla 8.15 se informa de una evaluación de sólo 300 diálogos, dado que el experimento sin usar GS sólo realizó 386 diálogos (a causa del fallo de los módulos GD en el diálogo nº 387). Por tanto, además de las peores estadísticas obtenidas al prescindir de la GS, también se produce el fallo de los gestores, con cancelación del bucle de diálogos en curso.

diálogo MD, el módulo de comprensión (Segarra et al., 2002), con simulación de errores, y la estrategia con verificación de confirmaciones implícitas en el simulador de usuario.

	Sin poda	Con poda
Nº de diálogos	300	300
Tasa de éxito	78,33%	91,67%
Nº medio errores por diálogo	2,08	2,11
% diálogos con introducción de errores	97,67%	98,00%
Tasa de éxito (en diálogos con errores)	76,50%	89,84%
Nº medio errores por diálogo (en diálogos con errores)	2,03	2,07
Nº medio turnos por diálogo	2,64	3,90
Nº total de turnos	793	1.169
Nº total de turnos a estados de RESPUESTA	732	710
Nº total de turnos a estados de CONFIRMACIÓN y/o PREGUNTA	61	416
Nº total de turnos a estados de ESPERA y de NO-ENTENDIDO	0	31
Nº total de turnos a estados de NUEVA-CONSULTA	0	12
Nº total de llamadas a MD-aux	0	46

Tabla 8.16: Comparativa sobre poda de transiciones según reglas de consistencia con el RH

Como se aprecia, la contribución de las técnicas de poda de transiciones según reglas de consistencia con el RH también es decisiva en el desarrollo de los diálogos. En las mismas condiciones experimentales, cuando no se podan transiciones se obtiene una tasa de éxito 13,34% menor² (desciende de 89,84% a 76,50%) y una duración media de los diálogos 1,26 turnos menor (desciende de 3,90 a 2,64 turnos).

Esta reducción de la duración de los diálogos se debe, fundamentalmente, a que el gestor de diálogo apenas transita a estados de confirmación o de pregunta. Como el simulador de usuario casi siempre facilita los atributos necesarios en su primera intervención, y en ausencia de poda de transiciones (que impedían transitar a estados de respuestas hasta conocer con suficiente confianza los valores de los atributos de consulta), el gestor de diálogo elige las transiciones a estados de respuesta. Por último, observar que si no se restringen las transiciones en el modelo, no se necesita el modelo auxiliar, MD-aux.

² Es probable que la tasa de éxito sea, realmente, bastante más baja. En esta experimentación, el GDS contesta al usuario ya en el turno S0, y, por otra parte, dado que la simulación de errores se concentra en los turnos iniciales de los diálogos, es muy probable que en el turno U0 se presente al menos un error, siendo incorrecta la respuesta en S0. Sin embargo, si S0 va acompañado de una confirmación implícita, el simulador de usuario tiene la oportunidad de corregir el error, en el turno U1, cuando pregunta por el siguiente objetivo. Como la verificación del éxito del diálogo, por comparación de RH, se realiza al final del diálogo, se contabilizaría como éxito, cuando sólo sería un éxito parcial.

8.7 Apunte sobre evaluación del gestor de diálogo y los modelos MD y MD*^e

Para evaluar el comportamiento (ante usuarios reales) del gestor de diálogo, cuando usa diferentes versiones del modelo de diálogo, se ha realizado una adquisición de diálogos donde el gestor de diálogo funcionaba con MD o con MD*^e. En esta experimentación, se han usado los mismos 14 escenarios considerados en el entrenamiento de los modelos mediante la técnica de simulación de usuarios (véase la Sección 7). En la Tabla 8.17 se resumen los resultados obtenidos.

	MD	MD*^e
Nº de diálogos	84	84
Tasa de éxito	73,81%	70,24%
Nº medio errores por diálogo	1,61	1,55
% diálogos con introducción de errores	86,90%	82,14%
Tasa de éxito (en diálogos con errores)	69,86%	63,77%
Nº medio errores por diálogo (en diálogos con errores)	1,85	1,88
Tasa de éxito normalizada (en diálogos con errores)	69,86%	64,97%
Nº medio turnos por diálogo	5,38	5,33
Nº total de turnos	452	448
Nº total de turnos a estados de RESPUESTA	205	207
Nº total de turnos a estados de CONFIRMACIÓN y/o PREGUNTA	234	238
Nº total de turnos a estados de ESPERA y de NO-ENTENDIDO	3	0
Nº total de turnos a estados de NUEVA-CONSULTA	0	1
Nº total de turnos a otros estados	10	2
Nº total de llamadas a MD-aux	11	3

Tabla 8.17: Evaluación con usuario real, comparativa de MD y MD*^e

Si se normaliza la tasa de diálogos con errores, el MD*^e alcanza una tasa de éxito (en diálogos con errores) 4,88% inferior a la obtenida con MD. La duración del diálogo es prácticamente la misma en ambos casos. Además, MD necesita llamar a MD-aux en un 2,43% de turnos frente a un 0,67% de turnos en MD*^e.

Estos resultados indican un mejor funcionamiento del modelo inicial, MD, aunque la experimentación es demasiado escasa (84 diálogos, con cada modelo, realizados por un solo usuario) para aventurar conclusiones definitivas al respecto.³

En esta misma experimentación, se recopiló información sobre los métodos de generalización semántica y las correspondientes transiciones a estados de usuario en los modelos de diálogo.

³ Por otra parte, esta experimentación se llevó a cabo antes de algunas modificaciones, significativas, en el gestor de diálogo, relativas a la traducción de frames a actos de diálogo (punto de partida de la generalización semántica).

Dado que las intervenciones del usuario son bastante breves, concisas, la generalización semántica frecuentemente es trivial. De este modo, el método y criterio más prioritarios (el método de concatenación condicionada a medidas de confianza y el criterio de comparación exacta) de la generalización semántica determinaron el siguiente estado de usuario en un 97,2% de las transiciones, en el MD, y en un 96,4% de las transiciones, en el MD*^e.

Para cada modelo, en las Tablas 8.18 y 8.19, se especifican el número de coincidencias encontradas entre descriptores de estados de usuario, disponibles en el modelo, y descriptores de actos de diálogo, generados usando los distintos métodos de generalización semántica y considerando los distintos criterios de correspondencia.

<i>Generalización semántica y modelo MD</i>	<i>Criterios de correspondencia</i>		
	Comparación exacta	Inclusión completa	Coincidencia e inclusión
<i>Métodos de generalización</i>			
Concatenación incondicional	523	1.332	929
Concatenación condicionada medidas confianza	518	1.315	---
Integración	524	1.396	872
Fragmentación aleatoria	533	1.409	645
Fragmentación condicionada medidas confianza	524	1.396	---
TOTAL	2.622	6.848	2.446

Tabla 8.18: Generalización semántica y modelo MD

<i>Generalización semántica y modelo MD*^e</i>	<i>Criterios de correspondencia</i>		
	Comparación exacta	Inclusión completa	Coincidencia e inclusión
<i>Métodos de generalización</i>			
Concatenación incondicional	505	1.264	871
Concatenación condicionada medidas confianza	498	1.235	---
Integración	508	1.342	784
Fragmentación aleatoria	515	1.354	637
Fragmentación condicionada medidas confianza	508	1.342	---
TOTAL	2.534	6.537	2.292

Tabla 8.19: Generalización semántica y modelo MD*^e

Se recopilaron 452 turnos en una serie de 84 diálogos y 448 turnos en la otra serie. En cada turno de diálogo se produce una transición a estado de usuario. En ausencia de generalización semántica, se hubiera dispuesto de, respectivamente, 452 y 448 descriptores de actos de diálogo para buscar siguiente estado (correspondencia biunívoca).

Al usar la generalización semántica, se dispone de un número de alternativas muy superior, aunque no tanto como indican las tablas anteriores, ya que:

- ❑ Las coincidencias por inclusión completa incluyen las coincidencias por comparación exacta.

- Distintos métodos de generalización semántica pueden generar el mismo descriptor de acto de diálogo (véase, por ejemplo, la Figura 5.6 donde se generan 15 descriptores pero sólo 8 son distintos).

Suponiendo que, en promedio, cada descriptor es generado por dos métodos de generalización, se puede calcular el número aproximado de descriptores de actos de diálogo, distintos, disponibles por transición:

$$\text{MD} \rightarrow \left(\frac{1}{2} \right) \cdot 6.848 \text{ descriptores} / 452 \text{ transiciones} = 7,58$$

$$\text{MD}^{*c} \rightarrow \left(\frac{1}{2} \right) \cdot 6.537 \text{ descriptores} / 448 \text{ transiciones} = 7,30$$

Como se observa, la técnica de generalización semántica permite multiplicar por 7 las posibilidades de transición ordinaria en el modelo (posibilidad de encontrar, en el modelo específico de diálogo, un estado de usuario adecuado a la entrada recibida, es decir, a los frames de usuario, y sin necesidad de recurrir a la transición extraordinaria basada en el modelo genérico de diálogo, MD-aux, y el registro histórico, RH).

Evaluación con usuario real del modelo MD	Nº total	%	Nº medio por diálogo
Diálogos	84		
Diálogos sin introducción de errores	11	13,10%	
Diálogos con introducción de errores	73	86,90%	
Diálogos concluidos con éxito	62	73,81%	
Diálogos con errores concluidos con éxito	51	69,86%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	135		1,61
Errores en atributo ORIGEN	40		0,48
Errores en atributo DESTINO	55		0,65
Errores en atributo FECHA-SALIDA	40		0,48
Turnos	452		5,38
Turnos (sistema) de Respuesta	205		2,44
Turnos (sistema) de Confirmación y Pregunta	12		0,14
Turnos (sistema) de Confirmación	153		1,82
Turnos (sistema) de Pregunta	69		0,82
Turnos (sistema) de Confirmación y Espera	3		0,04
Turnos (sistema) de sólo Nueva-Consulta	0		0
Turnos (sistema) de sólo No-Entendido	0		0
Otros Turnos (sistema)	10		0,12
Transiciones ordinarias a estados usuario	535	99,81%	
Transiciones extraordinarias a estados usuario	1	0,09%	
Transiciones ordinarias a estados sistema	526	98,13%	
Transiciones extraordinarias a estados sistema	10	1,87%	

Tabla 8.17a: Evaluación con usuario real del modelo MD

Evaluación con usuario real del modelo MD* ^e	Nº total	%	Nº medio por diálogo
Diálogos	84		
Diálogos sin introducción de errores	15	17,86%	
Diálogos con introducción de errores	69	82,14%	
Diálogos concluidos con éxito	59	70,24%	
Diálogos con errores concluidos con éxito	44	63,77%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	130		1,55
Errores en atributo ORIGEN	34		0,40
Errores en atributo DESTINO	53		0,63
Errores en atributo FECHA-SALIDA	43		0,52
Turnos	448		5,33
Turnos (sistema) de Respuesta	207		2,46
Turnos (sistema) de Confirmación y Pregunta	5		0,06
Turnos (sistema) de Confirmación	147		1,75
Turnos (sistema) de Pregunta	86		1,02
Turnos (sistema) de Confirmación y Espera	0		0
Turnos (sistema) de sólo Nueva-Consulta	1		0,01
Turnos (sistema) de sólo No-Entendido	0		0
Otros Turnos (sistema)	2		0,02
Transiciones ordinarias a estados usuario	531	99,81%	
Transiciones extraordinarias a estados usuario	1	0,19%	
Transiciones ordinarias a estados sistema	530	99,62%	
Transiciones extraordinarias a estados sistema	2	0,38%	

Tabla 8.17b: Evaluación con usuario real del modelo MD*^e

9. Conclusiones

9. Conclusiones

En la presente tesis, además de exponer el estado actual de las investigaciones relativas al desarrollo de sistemas de diálogo, se estableció como objetivo el desarrollo de varios módulos (en particular, el módulo gestor de diálogo) de un sistema de diálogo diseñado para la tarea del proyecto BASURDE. En las secciones precedentes de esta tesis se ha descrito el trabajo realizado y los logros alcanzados, disponiendo actualmente de un prototipo de sistema de diálogo, prototipo que se encuentra operativo y en el que se han integrado los módulos desarrollados en esta tesis.

En un sistema de diálogo, el módulo más importante en cuanto a supervisión de los diálogos en curso y determinación del comportamiento (respuestas, acciones, etc.) del sistema es el módulo gestor de diálogo. Tradicionalmente dicho módulo ha sido considerado desde una aproximación basada en reglas que incorporaban el conocimiento lingüístico de expertos en la tarea modelada (conocimiento de la semántica de la tarea, conocimiento de la pragmática del diálogo, etc.).

En esta tesis se ha considerado la aproximación estadística, que es más innovadora que las aproximaciones basadas en el uso de reglas. Por tanto, se obtuvieron modelos estocásticos de diálogo a partir de un corpus de diálogos etiquetados de la tarea. Sin embargo, dado el reducido tamaño del corpus de BASURDE, los modelos inferidos tienen una cobertura parcial del espacio de sucesos propio de la tarea. Así, ciertas situaciones de diálogos están adecuadamente representadas en los modelos, mientras que otras situaciones están deficientemente modeladas (por escasez de muestras en el corpus) o incluso no constan en los modelos (por ausencia de muestras).

Ante esta problemática, se han planteado varias soluciones, integrándose todas ellas en el gestor de diálogo desarrollado. Los modelos estocásticos de diálogo considerados son modelos de n-gramas de actos de diálogo. Dado el tamaño y características del corpus, se han inferido sólo modelos de bigramas y, además, excluyendo algún nivel de etiquetado. Se trata, por tanto, de modelos sencillos donde sólo se han incluido los estados de diálogos y las transiciones entre estados que estuvieran presentes en el corpus con una base estadística significativa o, al menos, aceptable.

Sin embargo, usando solamente estos modelos, el gestor de diálogo puede seleccionar estrategias de diálogo correctas para bastantes situaciones de diálogo, pero no para cualquier posible diálogo en el ámbito de la tarea. En consecuencia, fue necesario desarrollar otros procedimientos que complementarían a los modelos estocásticos, en particular: la generalización semántica, el uso del registro histórico y el uso de las medidas de confianza.

La generalización semántica consiste en un suavizado de los frames de usuario recibidos y facilita la adecuada transición a estados de usuario en los modelos estocásticos. Como se vio en la evaluación del sistema, el uso de la generalización semántica en el gestor de diálogo supone mejorar, aproximadamente, un 23% la tasa de éxito de los diálogos.

Mediante el registro histórico se tiene en cuenta la historia del diálogo en curso (información que no consta en el estado actual de diálogo en un modelo de bigramas) y se puede implementar un conjunto de reglas que garanticen la coherencia de las respuestas del sistema con la historia del diálogo (al orientar la selección de los nuevos estados de sistema por

medio de la poda de las transiciones a estados que no sean consistentes con los turnos previos del diálogo en curso). De nuevo, en la evaluación del sistema se vio la importancia de incorporar el registro histórico (y estas reglas de consistencia con su contenido) en el gestor de diálogo puesto que permite mejorar la tasa de éxito en torno a un 13%.

Por último, y con el mismo fin de facilitar una más adecuada transición a estados de sistema en los modelos estocásticos, se introdujo el uso de las medidas de confianza (procedentes de otros módulos, reconocimiento del habla y/o comprensión del lenguaje, y almacenadas en el registro histórico del gestor de diálogo). La mejora introducida con su uso, tal como se indica en la evaluación del sistema, es también significativa: la tasa de éxito asciende en torno a un 25% o un 30%, dependiendo de las condiciones experimentales.

En resumen, se ha desarrollado un módulo gestor de diálogo que aplica una estrategia de diálogo híbrida en el sentido de que está determinada, en parte, por los modelos estocásticos de diálogo y, en parte, por unas reglas de consistencia con la historia del diálogo (reglas que tienen en cuenta el conocimiento o desconocimiento de los valores de los atributos de la tarea y sus medidas de confianza). Los resultados experimentales, tanto con entrada textual como con entrada hablada, muestran el correcto funcionamiento del sistema de diálogo cuando se usa este gestor de diálogo.

Como objetivos secundarios de esta tesis, se han desarrollado sendos módulos para la comprensión del lenguaje natural y para la generación de respuestas en lenguaje natural.

El desarrollo del generador de respuestas era una tarea obligada por cuanto, en su momento, no se disponía de ningún módulo que realizara tal función en el sistema de diálogo del proyecto BASURDE. Aunque algunos grupos de investigación han planteado aproximaciones estadísticas al problema de la generación de respuestas, en esta tesis se optó, dado el carácter auxiliar de este módulo respecto al módulo gestor de diálogo, por una solución más sencilla consistente en el uso de plantillas y de reglas para su selección y rellenado. Tuvo, además, que definirse un formato para el intercambio de información entre los dos módulos, gestor de diálogo y generador de respuestas, es decir, definirse el formato de lo que llamamos frames del sistema.

En cuanto al desarrollo del módulo de comprensión del lenguaje natural, hay que recordar que no era una tarea imprescindible dado que ya existía un módulo de comprensión para el proyecto BASURDE, resultado del trabajo de otros investigadores del mismo grupo. Sin embargo, se juzgó interesante explorar las posibilidades de otra solución al problema, solución en parte estadística (en la determinación de los conceptos de la tarea) y en parte basada en reglas (en la identificación de los atributos de la tarea). En la correspondiente sección de la tesis, además de describir este módulo, se informó de su evaluación y se adelantaron unas conclusiones sobre el mismo. Aquí, tan sólo, corresponde resaltar el funcionamiento aceptable de este nuevo módulo de comprensión, su acertada traducción a frames (particularmente, en frases de longitud corta o media), y sus mejores prestaciones, en cuanto a velocidad, cuando funciona integrado con el resto del sistema de diálogo y en el modo de simulación de usuarios.

La consideración de las posibles alternativas para mejorar el funcionamiento del gestor de diálogo, en particular las deficiencias presentes en los modelos estocásticos de diálogo, derivadas de la escasez de muestras en el corpus de aprendizaje, llevaron a plantear el desarrollo del simulador de usuarios. De este modo, se podrían adquirir nuevos diálogos (sin apenas coste, si se compara con la adquisición con usuarios reales) y usar este corpus de

diálogos sintéticos para obtener nuevos modelos estocásticos o, más bien, mejorar o reentrenar los modelos existentes.

Para que este nuevo objetivo se cumpla, es esencial que el simulador de usuario actúe de la manera más parecida posible a como lo harían los usuarios reales. El módulo principal de nuestro simulador de usuario es el gestor de diálogo del usuario (GDU). Este módulo es una versión modificada del gestor de diálogo del sistema (GDS). Como el GDS, el GDU usa los modelos estocásticos de diálogo inferidos a partir del corpus de BASURDE. En consecuencia, parte del “comportamiento” del GDU queda determinado, estadísticamente, por los comportamientos reales de los usuarios que participaron en la adquisición del corpus de BASURDE. Además, la estrategia de diálogo del GDU se ha completado con la definición de una serie de reglas para la selección de los nuevos estados de usuario. Estas reglas tienen en cuenta los objetivos a alcanzar en los escenarios objeto de simulación y buscan transiciones a estados donde se pregunte por los objetivos y/o se responda a las preguntas o confirmaciones que el sistema hubiera planteado previamente. Es decir, el GDU, con estas reglas, implementa una estrategia de diálogo colaborativa con el sistema. El GDU representaría a un usuario genérico dispuesto a colaborar con el sistema que le atiende en la obtención de sus objetivos.

En la experimentación usando el simulador de usuario, se considera que los diálogos que concluyen con éxito (es decir, suministrando al simulador la información que solicitó) son diálogos en los que el GDS siguió una estrategia correcta, eligió las transiciones adecuadas en los modelos estocásticos de diálogo. En consecuencia, tal estrategia de diálogo debería ser primada para facilitar su selección en diálogos posteriores. Para ello, se modificó el GDS de modo que fuera capaz de detectar el éxito de las simulaciones (mediante la comparación de los respectivos registros históricos) y que pudiera modificar el modelo estocástico de diálogo, incrementando las probabilidades de las transiciones elegidas en las simulaciones concluidas con éxito.

Los resultados experimentales muestran la viabilidad de la técnica propuesta. Se ha llevado a cabo el reaprendizaje o adaptación del modelo estocástico inicial (inferido a partir del corpus de BASURDE), mediante la generación de series de 5.000 diálogos simulados, en diferentes situaciones (ausencia y presencia de errores en los frames generados por los módulos de comprensión). En todas las situaciones, se han obtenido nuevos modelos estocásticos en los que las probabilidades de las transiciones se han reajustado en función del resultado de las simulaciones. Se ha observado, además, que la modificación de las probabilidades es convergente (se alcanza la convergencia después de ejecutar, aproximadamente, 1.000 simulaciones que concluyan con éxito).

La evaluación comparativa de los diferentes modelos estocásticos de diálogo (el modelo inicial y los modelos reentrenados mediante simulación de usuario) proporciona unos resultados positivos aunque modestos. La tasa de éxito de diálogo mejora en torno a un 1% o un 2% cuando se usan los modelos reentrenados. La duración de los diálogos tiende a disminuir, obteniéndose con los nuevos modelos un número medio de turnos por diálogo en torno a un 0,1 o un 0,2 menor que cuando se usa el modelo inicial. Además, al usar los nuevos modelos, es importante apreciar la disminución de las transiciones a ciertos estados de sistema (estados de espera, de no entendido, o de sólo nueva consulta), estados que no son bien recibidos por los usuarios reales y que no facilitan el adecuado progreso del diálogo, así como la disminución de las llamadas al modelo estocástico auxiliar (dado que este modelo no suele facilitar las transiciones más adecuadas sino, tan sólo, transiciones que permi-

ten la continuidad del diálogo, es decir, que eviten el bloqueo del sistema). Estos aspectos apuntan a que se mejorará la “calidad” de los diálogos llevados a cabo con usuarios reales cuando se usen los nuevos modelos.

Como trabajos futuros que den continuidad a lo realizado en la presente tesis se apuntan dos líneas principales: la adaptación del gestor de diálogo para utilizar modelos estocásticos inferidos a partir del corpus del proyecto DIHANA, y la modificación y mejora del simulador de usuario en función de la evaluación de los nuevos modelos en diálogos con usuarios reales. También pueden considerarse otros objetivos: adaptar el módulo de comprensión usando modelos inferidos del corpus de DIHANA, y reconsiderar el módulo generador de respuestas desde una perspectiva más estadística.

Dado que los modelos estocásticos de diálogo usados hasta ahora en el gestor de diálogo tenían su debilidad en el reducido tamaño del corpus de aprendizaje (los 227 diálogos del corpus del proyecto BASURDE), podemos confiar en que los modelos que se infieran a partir de un corpus de mayor tamaño serán claramente superiores. En esta línea, el reciente etiquetado en términos de actos de diálogo de los 900 diálogos del corpus del proyecto DIHANA nos permite aprender automáticamente nuevos modelos estocásticos de diálogo que, sin duda, tendrán una cobertura más amplia del espacio de sucesos de la tarea que los inferidos a partir del corpus de BASURDE. De hecho, tales modelos inferidos a partir del corpus de DIHANA ya existen, aunque todavía no se han probado al ser necesarias algunas modificaciones en el módulo gestor de diálogo, derivadas de algunas diferencias en las etiquetas y reglas de etiquetado usadas en los corpus de BASURDE y DIHANA.

Una vez se integren los modelos estocásticos de diálogo inferidos a partir del corpus de DIHANA en el sistema de diálogo podrá determinarse si conviene ampliar el rango de los n-gramas (por ejemplo, de bigramas a trigramas de actos de diálogo) o incluir todos los niveles de etiquetado (es decir, considerar en los descriptores de los estados de los modelos el 3º nivel de etiquetado, nivel que fue excluido en los modelos de BASURDE). Igualmente, entonces, se podrá observar el nuevo peso que en la determinación de la estrategia de diálogo tengan los procedimientos complementarios (como la generalización semántica, las reglas de consistencia con el registro histórico, las llamadas a modelos auxiliares, genéricos, etc.). Dado que los modelos de DIHANA ofrecerán una mejor cobertura del espacio de sucesos de la tarea, cabe anticipar, con cierta seguridad, que la nueva estrategia de diálogo será más estocástica (con más peso en el uso de los modelos estocásticos de diálogo) y que, como la cobertura del espacio de sucesos seguirá sin ser completa, convendrá mantener los procedimientos complementarios (los mismos desarrollados en esta tesis o, tal vez, otros) para resolver las situaciones excepcionales que se presenten en el curso de los diálogos.

En cuanto al trabajo futuro en la modificación y mejora del simulador de usuario, indicar, en primer lugar, la conveniencia de proceder a una evaluación con usuarios reales de los nuevos modelos obtenidos en esta tesis, los modelos adaptados a partir del modelo inicial de BASURDE. Esta evaluación pondrá de manifiesto hasta qué punto los nuevos modelos proporcionan una mejor estrategia, y también indicará los puntos débiles de la técnica de simulación (es decir, las diferencias entre el comportamiento del simulador y los comportamientos de los usuarios reales). De este modo, podrán introducirse, en los módulos del simulador de usuario, las modificaciones que sean pertinentes.

Por último, aunque los modelos iniciales de DIHANA sean superiores a los de BASURDE, nada nos impide tratar de mejorarlos mediante la misma técnica de adaptación a través de

series de diálogos simulados que se ha seguido en esta tesis. Así pues, en la lista de trabajos futuros hay que incluir el reaprendizaje mediante simulación de los modelos estocásticos de diálogo inferidos a partir del corpus de DIHANA, y la evaluación de los nuevos modelos que se obtengan preferentemente con usuarios reales.

10. Anexos

10. Anexos

10.1 Generador de respuestas

En la siguiente Tabla 10.1 se muestran los frames de sistema generados en cada uno de los estados de sistema existentes en el modelo de diálogo.

	<i>Estado del modelo de diálogo</i>	<i>Frames de sistema</i>
1	(S:Apertura:Nil)	(APERTURA) 1.00
12	(S:Cierre:Nil)	(CIERRE) 1.00
51	(S:Confirmacion:Destino)	(CONFIRMACION) 1.00 DESTINO:ccc d.dd
228	(S:Confirmacion:Destino) (S:Pregunta:Origen)	(CONFIRMACION) 1.00 DESTINO:ccc d.dd (SOLICITUD) 1.00 ORIGEN:ccc d.dd
21	(S:Confirmacion:Destino,Dia_salida,Origen)	(CONFIRMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd FECHA-SALIDA:dd-dd-dddd d.dd
13	(S:Confirmacion:Destino,Origen)	(CONFIRMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd
218	(S:Confirmacion:Destino,Origen) (S:Pregunta:Dia_salida)	(CONFIRMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd (SOLICITUD) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd
4	(S:Confirmacion:Dia_salida)	(CONFIRMACION) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd
130	(S:Confirmacion:Dia_salida,Origen)	(CONFIRMACION) 1.00 ORIGEN:ccc d.dd FECHA-SALIDA:dd-dd-dddd d.dd
46	(S:Confirmacion:Hora_llegada) 2 posibilidades, según en el HR se guarde, o no, una hora de llegada además, puede ser: HORA-LLEGADA:??? 0.00	(CONFIRMACION) 1.00 { HORA-LLEGADA:dd.dd 1.00 }
221	(S:Confirmacion:Hora_llegada) (S:Espera:Nil)	(CONFIRMACION) 1.00 { HORA-LLEGADA:dd.dd 1.00 } (ESPERA) 1.00
53	(S:Confirmacion:Hora_llegada) (S:Espera:Nil) (S:Respuesta:Hora_llegada)	(CONFIRMACION) 1.00 HORA-LLEGADA:??? 1.00 (ESPERA) 1.00

	(S:Nueva_consulta:Nil)	(INFORMACION) 1.00 HORA-LLEGADA:dd.dd 0.00 (NUEVA-CONSULTA) 1.00
286	(S:Confirmacion:Hora_llegada,Hora_salida) (S:Espera:Nil) (S:Respuesta:Hora_llegada,Hora_salida) (S:Nueva_consulta:Nil)	(CONFIRMACION) 1.00 HORA-SALIDA:??? 1.00 HORA-LLEGADA:??? 1.00 (ESPERA) 1.00 (INFORMACION) 1.00 HORA-SALIDA:dd.dd 1.00 HORA-LLEGADA:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
66	(S:Confirmacion:Hora_salida)	(CONFIRMACION) 1.00 { HORA-SALIDA:dd.dd 1.00 }
59	(S:Confirmacion:Hora_salida) (S:Espera:Nil)	(CONFIRMACION) 1.00 { HORA-SALIDA:dd.dd 1.00 } (ESPERA) 1.00
23	(S:Confirmacion:Hora_salida) (S:Espera:Nil) (S:Pregunta:Tipo_tren)	(CONFIRMACION) 1.00 { HORA-SALIDA:dd.dd 1.00 } (ESPERA) 1.00 (SOLICITUD) 1.00 TIPO-TREN:ccc d.dd
3	(S:Confirmacion:Hora_salida) (S:Espera:Nil) (S:Respuesta:Hora_salida) (S:Nueva_consulta:Nil)	(CONFIRMACION) 1.00 HORA-SALIDA:??? 1.00 (ESPERA) 1.00 (INFORMACION) 1.00 HORA-SALIDA:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
174	(S:Confirmacion:Hora_salida) (S:Espera:Nil) (S:Respuesta:Hora_salida) (S:Pregunta:Tipo_tren)	(CONFIRMACION) 1.00 HORA-SALIDA:??? 1.00 (ESPERA) 1.00 (INFORMACION) 1.00 HORA-SALIDA:dd.dd 1.00 (SOLICITUD) 1.00 TIPO-TREN:ccc d.dd
88	(S:Confirmacion:Hora_salida_v)	(CONFIRMACION) 1.00 HORA-SALIDA-V:??? 1.00
103	(S:Confirmacion:Hora_salida_v) (S:Espera:Nil)	(CONFIRMACION) 1.00 HORA-SALIDA-V:??? 1.00 (ESPERA) 1.00
16	(S:Confirmacion:Hora_salida_v) (S:Espera:Nil) (S:Respuesta:Hora_salida_v) (S:Nueva_consulta:Nil)	(CONFIRMACION) 1.00 HORA-SALIDA-V:??? 1.00 (ESPERA) 1.00 (INFORMACION) 1.00 HORA-SALIDA-V:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
30	(S:Confirmacion:Origen)	(CONFIRMACION) 1.00 ORIGEN:ccc d.dd
93	(S:Confirmacion:Precio) (S:Espera:Nil) (S:Pregunta:Tipo_tren)	(CONFIRMACION) 1.00 PRECIO:??? 1.00 (ESPERA) 1.00

		(SOLICITUD) 1.00 TIPO-TREN:ccc d.dd
140	(S:Confirmacion:Precio) (S:Espera:Nil) (S:Respuesta:Precio) (S:Nueva_consulta:Nil)	(CONFIRMACION) 1.00 PRECIO:??? 0.00 (ESPERA) 1.00 (INFORMACION) 1.00 PRECIO:dd 1.00 (NUEVA-CONSULTA) 1.00
32	(S:Confirmacion:Precio) (S:Espera:Nil) (S:Respuesta:Precio) (S:Pregunta:Hora_salida)	(CONFIRMACION) 1.00 PRECIO:??? 1.00 (ESPERA) 1.00 (INFORMACION) 1.00 PRECIO:dd 1.00 (SOLICITUD) 1.00
132	(S:Confirmacion:Precio) (S:Espera:Nil) (S:Respuesta:Precio) (S:Pregunta:Tipo_tren)	(CONFIRMACION) 1.00 PRECIO:??? 1.00 (ESPERA) 1.00 (INFORMACION) 1.00 PRECIO:dd 1.00 (SOLICITUD) 1.00 TIPO-TREN:ccc d.dd
136	(S:Confirmacion:Servicio)	(CONFIRMACION) 1.00 { SERVICIO:ccc d.dd }
124	(S:Confirmacion:Tipo_tren)	(CONFIRMACION) 1.00 { TIPO-TREN:ccc d.dd }
243	(S:Espera:Nil) (S:Pregunta:Servicio)	(ESPERA) 1.00 (SOLICITUD) 1.00 SERVICIO:ccc d.dd
60	(S:Espera:Nil) (S:Pregunta:Tipo_tren)	(ESPERA) 1.00 (SOLICITUD) 1.00 TIPO-TREN:ccc d.dd
27	(S:Espera:Nil) (S:Respuesta:Hora_llegada) (S:Nueva_consulta:Nil)	(ESPERA) 1.00 (INFORMACION) 1.00 HORA-LLEGADA:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
262	(S:Espera:Nil) (S:Respuesta:Hora_llegada,Hora_salida) (S:Nueva_consulta:Nil)	(ESPERA) 1.00 (INFORMACION) 1.00 HORA-SALIDA:dd.dd 1.00 HORA-LLEGADA:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
48	(S:Espera:Nil) (S:Respuesta:Hora_llegada_v) (S:Nueva_consulta:Nil)	(ESPERA) 1.00 (INFORMACION) 1.00 HORA-LLEGADA-V:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
214	(S:Espera:Nil) (S:Respuesta:Hora_llegada_v,Tipo_tren) (S:Nueva_consulta:Nil)	(ESPERA) 1.00 (INFORMACION) 1.00 TIPO-TREN:ccc 1.00 HORA-LLEGADA-V:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
44	(S:Espera:Nil) (S:Respuesta:Hora_salida)	(ESPERA) 1.00 (INFORMACION) 1.00

	(S:Nueva_consulta:Nil)	HORA-SALIDA:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
126	(S:Espera:Nil) (S:Respuesta:Hora_salida) (S:Pregunta:Tipo_tren)	(ESPERA) 1.00 (INFORMACION) 1.00 HORA-SALIDA:dd.dd 1.00 (SOLICITUD) 1.00 TIPO-TREN:ccc d.dd
49	(S:Espera:Nil) (S:Respuesta:Hora_salida_v) (S:Nueva_consulta:Nil)	(ESPERA) 1.00 (INFORMACION) 1.00 HORA-SALIDA-V:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
10	(S:Espera:Nil) (S:Respuesta:Precio) (S:Nueva_consulta:Nil)	(ESPERA) 1.00 (INFORMACION) 1.00 PRECIO:dd 1.00 (NUEVA-CONSULTA) 1.00
267	(S:Espera:Nil) (S:Respuesta:Precio) (S:Pregunta:Servicio)	(ESPERA) 1.00 (INFORMACION) 1.00 PRECIO:dd 1.00 (SOLICITUD) 1.00 SERVICIO:ccc d.dd
81	(S:Espera:Nil) (S:Respuesta:Precio,Tipo_tren) (S:Nueva_consulta:Nil)	(ESPERA) 1.00 (INFORMACION) 1.00 TIPO-TREN:ccc 1.00 PRECIO:dd 1.00 (NUEVA-CONSULTA) 1.00
25	(S:Espera:Nil) (S:Respuesta:Servicio) (S:Nueva_consulta:Nil)	(ESPERA) 1.00 (INFORMACION) 1.00 SERVICIO:ccc 1.00 (NUEVA-CONSULTA) 1.00
7	(S:Espera:Nil) (S:Respuesta:Tipo_tren) (S:Nueva_consulta:Nil)	(ESPERA) 1.00 (INFORMACION) 1.00 TIPO-TREN:ccc 1.00 (NUEVA-CONSULTA) 1.00
98	(S:No_entendido:Nil)	(NO-ENTENDIDO) 1.00
279	(S:No_entendido:Nil) (S:Confirmacion:Destino) (S:No_entendido:Nil)	(NO-ENTENDIDO) 1.00 (CONFIRMACION) 1.00 DESTINO:ccc d.dd (NO-ENTENDIDO) 1.00
64	(S:No_entendido:Nil) (S:Confirmacion:Destino,Dia_salida) (S:No_entendido:Nil)	(NO-ENTENDIDO) 1.00 (CONFIRMACION) 1.00 DESTINO:ccc d.dd FECHA-SALIDA:dd-dd-dddd d.dd (NO-ENTENDIDO) 1.00
62	(S:No_entendido:Nil) (S:Confirmacion:Destino,Dia_salida,Origen) (S:No_entendido:Nil)	(NO-ENTENDIDO) 1.00 (CONFIRMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd FECHA-SALIDA:dd-dd-dddd d.dd (NO-ENTENDIDO) 1.00
92	(S:No_entendido:Nil) (S:Confirmacion:Destino,Origen)	(NO-ENTENDIDO) 1.00 (CONFIRMACION) 1.00

		ORIGEN:ccc d.dd DESTINO:ccc d.dd
41	(S:No_entendido:Nil) (S:Confirmacion:Destino,Origen) (S:No_entendido:Nil)	(NO-ENTENDIDO) 1.00 (CONFIRMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd (NO-ENTENDIDO) 1.00
123	(S:No_entendido:Nil) (S:Confirmacion:Destino,Origen,Tipo_tren) (S:No_entendido:Nil)	(NO-ENTENDIDO) 1.00 (CONFIRMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd TIPO-TREN:ccc d.dd (NO-ENTENDIDO) 1.00
147	(S:No_entendido:Nil) (S:Confirmacion:Origen) (S:No_entendido:Nil)	(NO-ENTENDIDO) 1.00 (CONFIRMACION) 1.00 ORIGEN:ccc d.dd (NO-ENTENDIDO) 1.00
244	(S:No_entendido:Nil) (S:Confirmacion:Tipo_tren)	(NO-ENTENDIDO) 1.00 (CONFIRMACION) 1.00 TIPO-TREN:ccc d.dd
141	(S:No_entendido:Nil) (S:Nueva_consulta:Nil)	(NO-ENTENDIDO) 1.00 (NUEVA-CONSULTA) 1.00
54	(S:No_entendido:Nil) (S:Pregunta:Dia_salida)	(NO-ENTENDIDO) 1.00 (SOLICITUD) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd
29	(S:Nueva_consulta:Nil)	(NUEVA-CONSULTA) 1.00
145	(S:Pregunta:Destino)	(SOLICITUD) 1.00 DESTINO:ccc d.dd
108	(S:Pregunta:Dia_salida)	(SOLICITUD) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd
127	(S:Pregunta:Hora_salida)	(SOLICITUD) 1.00 { HORA-SALIDA:dd.dd 1.00 }
178	(S:Pregunta:Origen)	(SOLICITUD) 1.00 ORIGEN:ccc d.dd
79	(S:Pregunta:Servicio)	(SOLICITUD) 1.00 SERVICIO:ccc d.dd
19	(S:Pregunta:Tipo_tren)	(SOLICITUD) 1.00 TIPO-TREN:ccc d.dd
37	(S:Respuesta:Hora_llegada) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 HORA-LLEGADA:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
35	(S:Respuesta:Hora_llegada,Hora_salida) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 HORA-SALIDA:dd.dd 1.00 HORA-LLEGADA:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
287	(S:Respuesta:Hora_llegada,Hora_salida) (S:Pregunta:Tipo_tren)	(INFORMACION) 1.00 HORA-SALIDA:dd.dd 1.00 HORA-LLEGADA:dd.dd 1.00 (SOLICITUD) 1.00 TIPO-TREN:ccc d.dd

170	(S:Respuesta:Hora_llegada_v) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 HORA-LLEGADA-V:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
105	(S:Respuesta:Hora_llegada_v,Hora_salida_v) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 HORA-SALIDA-V:dd.dd 1.00 HORA-LLEGADA-V:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
20	(S:Respuesta:Hora_salida) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 HORA-SALIDA:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
110	(S:Respuesta:Hora_salida) (S:Pregunta:Tipo_tren)	(INFORMACION) 1.00 HORA-SALIDA:dd.dd 1.00 (SOLICITUD) 1.00 TIPO-TREN:ccc d.dd
76	(S:Respuesta:Hora_salida_v) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 HORA-SALIDA-V:dd.dd 1.00 (NUEVA-CONSULTA) 1.00
17	(S:Respuesta:Precio) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 PRECIO:dd 1.00 (NUEVA-CONSULTA) 1.00
86	(S:Respuesta:Precio,Tipo_tren) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 TIPO-TREN:ccc 1.00 PRECIO:dd 1.00 (NUEVA-CONSULTA) 1.00
83	(S:Respuesta:Servicio) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 SERVICIO:ccc 1.00 (NUEVA-CONSULTA) 1.00
143	(S:Respuesta:Tiempo_recorrido) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 TIEMPO-RECORRIDO:dd 1.00 (NUEVA-CONSULTA) 1.00
119	(S:Respuesta:Tipo_tren) (S:Nueva_consulta:Nil)	(INFORMACION) 1.00 TIPO-TREN:ccc 1.00 (NUEVA-CONSULTA) 1.00

En la siguiente Tabla 10.2 se muestran las plantillas usadas para generar las frases de respuesta del sistema y sus correspondencias con los frames de sistema, recibidos desde el módulo gestor de diálogo. En la columna de las plantillas de frases, los nombres en cursiva y delimitados con los caracteres “<” y “>” representan nombres de variables atributo, que se sustituyen con la expresión, en lenguaje natural, de los correspondientes valores de atributos, cuando la plantilla se transforma en segmento de una frase de respuesta.

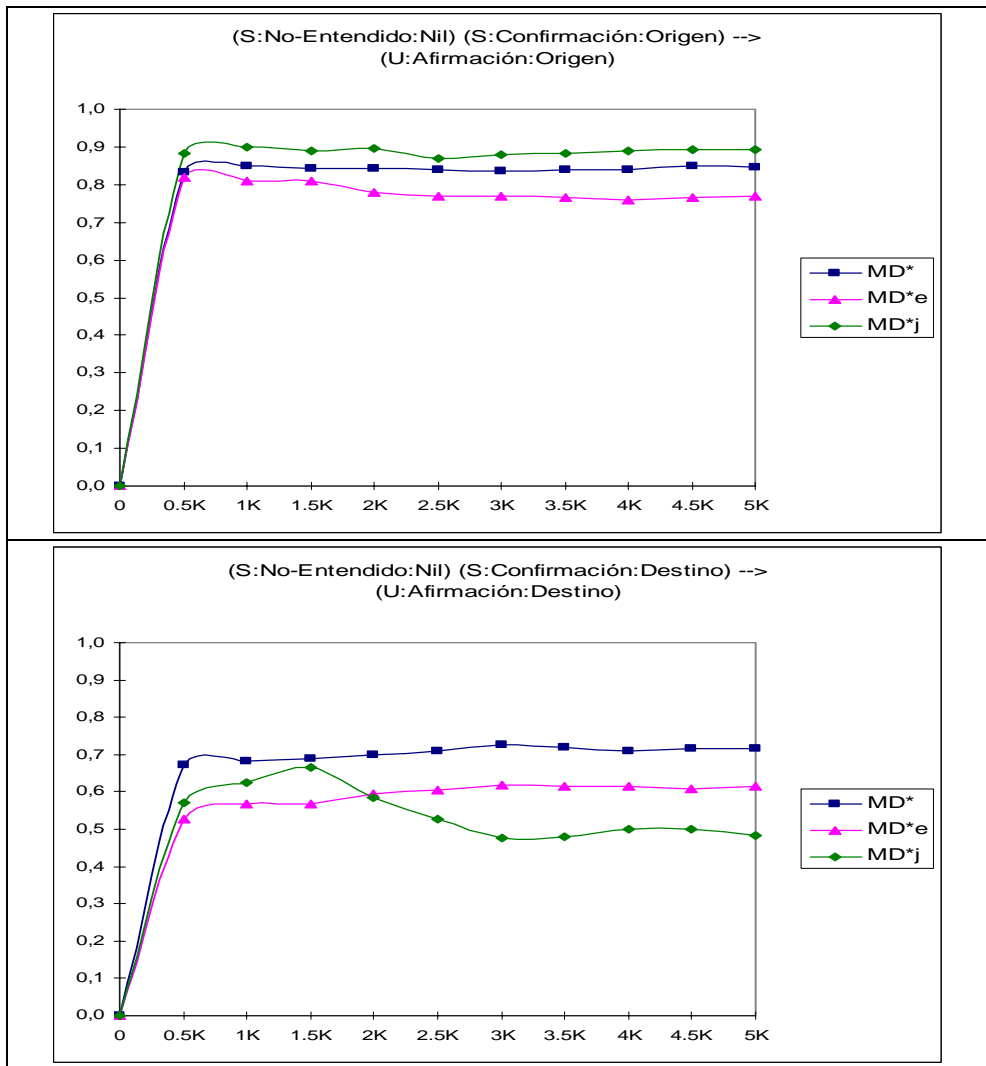
<i>Componentes de frames de sistema</i>	<i>Plantillas de frases</i>
(APERTURA)	bienvenido al servicio de información de trenes . ¿ en qué puedo ayudarle ?
(CIERRE)	gracias por utilizar este servicio . feliz viaje .
(NUEVA-CONSULTA)	¿ desea algo más ?
(NO-ENTENDIDO)	disculpe pero no le he entendido .

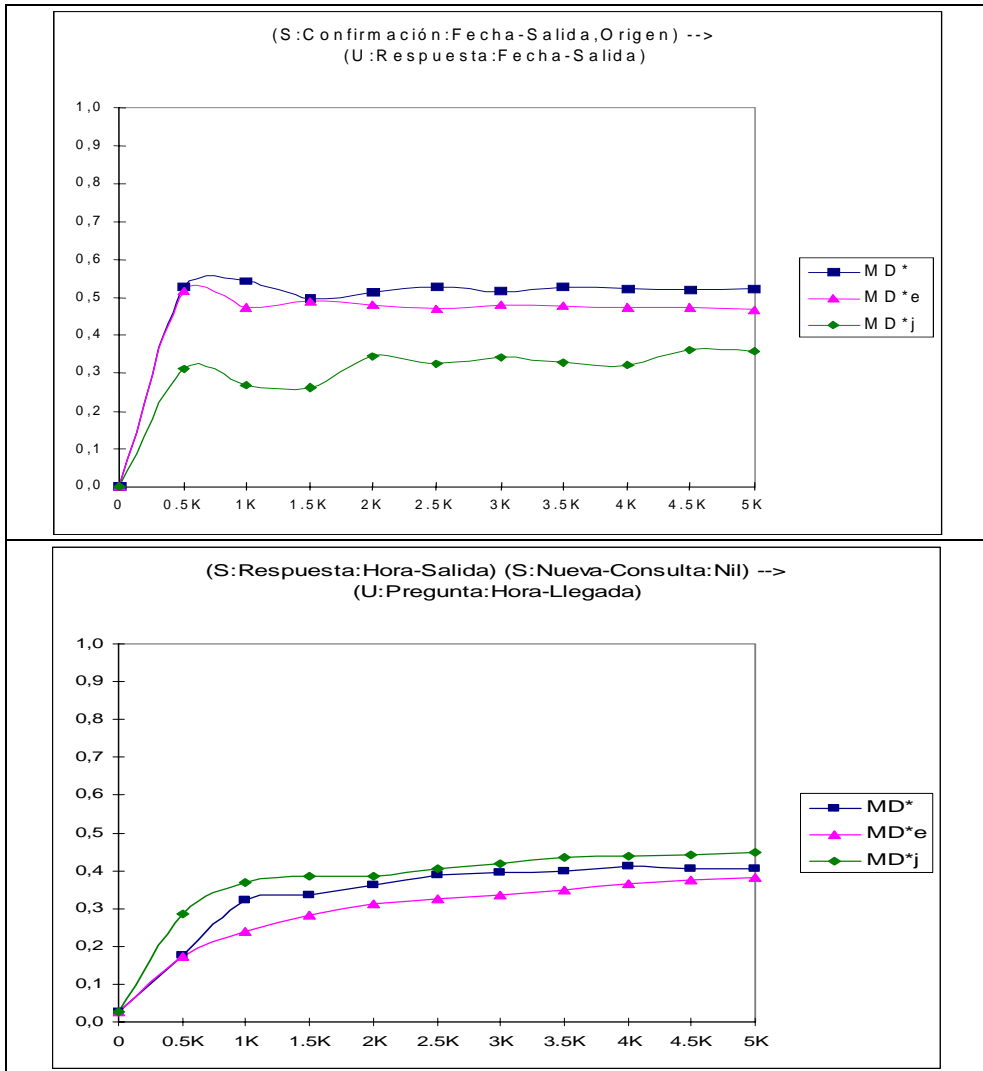
(ESPERA)	un momento por favor .
(CONFIRMACIÓN) ORIGEN	¿ quiere salir de <Origen> ?
(CONFIRMACIÓN) DESTINO	¿ quiere ir a <Destino> ?
(CONFIRMACIÓN) ORIGEN-V	¿ quiere salir de <Origen> ?
(CONFIRMACIÓN) DESTINO-V	¿ quiere volver a <Destino> ?
(CONFIRMACIÓN) FECHA-SALIDA	¿ quiere viajar el <Día-Salida> ?
(CONFIRMACIÓN) FECHA-SALIDA-V	¿ quiere volver el <Día-Salida> ?
(CONFIRMACIÓN) SERVICIOS	¿ quiere viajar en <Servicio> ?
(CONFIRMACIÓN) TIPO-TREN	¿ quiere viajar en <Tipo-Tren> ?
(CONFIRMACIÓN) HORA-LLEGADA	¿ quiere saber a qué hora llega ?
(CONFIRMACIÓN) HORA-SALIDA	¿ quiere saber a qué hora sale ?
(CONFIRMACIÓN) HORA-SALIDA-V	¿ quiere saber a qué hora sale el tren de vuelta ?
(CONFIRMACIÓN) ORIGEN DESTINO	¿ quiere ir de <Origen> a <Destino> ?
(CONFIRMACIÓN) ORIGEN-V DESTINO-V	¿ quiere volver de <Origen> a <Destino> ?
(CONFIRMACIÓN) DESTINO FECHA-SALIDA	¿ quiere ir a <Destino> el <Día-Salida> ?
(CONFIRMACIÓN) ORIGEN FECHA-SALIDA	¿ quiere viajar desde <Origen> el <Día-Salida> ?
(CONFIRMACIÓN) ORIGEN DESTINO FECHA-SALIDA	¿ quiere ir de <Origen> a <Destino> el <Día-Salida> ?
(CONFIRMACIÓN) ORIGEN DESTINO TIPO-TREN	¿ quiere ir de <Origen> a <Destino> en <Tipo-Tren> ?
(CONFIRMACIÓN) HORA-LLEGADA (ESPERA)	le consulto la hora de llegada .
(CONFIRMACIÓN) HORA-SALIDA (ESPERA)	le consulto la hora de salida .
(CONFIRMACIÓN) HORA-SALIDA-V (ESPERA)	le consulto la hora de salida del tren de vuelta .
(CONFIRMACIÓN) PRECIO (ESPERA)	le consulto el precio .
(CONFIRMACIÓN) HORA-SALIDA HORA-LLEGADA (ESPERA)	le consulto horarios de salida y de llegada .
(SOLICITUD) ORIGEN	¿ de dónde quiere salir ?
(SOLICITUD) DESTINO	¿ a dónde quiere ir ?
(SOLICITUD) FECHA-SALIDA	¿ qué día quiere salir ?
(SOLICITUD) TIPO-TREN	¿ en qué tipo de tren quiere viajar ?
(SOLICITUD) HORA-SALIDA	¿ quiere saber los horarios ?
(SOLICITUD) HORA-SALIDA-V	¿ quiere saber los horarios de la vuelta ?
(SOLICITUD) SERVICIOS	¿ quiere información sobre servicios ?
(INFORMACIÓN)	no hay trenes que cumplan su petición .
(INFORMACIÓN) HORA-SALIDA	el tren sale a <Hora-Salida> .
(INFORMACIÓN) HORA-LLEGADA	el tren llega a <Hora-Llegada> .
(INFORMACIÓN) PRECIO	el billete cuesta <Precio> euros .
(INFORMACIÓN) SERVICIOS	el tren dispone de <Servicio> .
(INFORMACIÓN) TIPO-TREN	es un tren <Tipo-Tren> .
(INFORMACIÓN) HORA-SALIDA-V	el tren de vuelta sale a <Hora-Salida> .
(INFORMACIÓN) HORA-LLEGADA-V	el tren de vuelta llega a <Hora-Llegada> .
(INFORMACIÓN) HORA-SALIDA HORA-LLEGADA	el tren sale a <Hora-Salida> y llega a <Hora-Llegada> .
(INFORMACIÓN) HORA-SALIDA-V HORA-LLEGADA-V	el tren de vuelta sale a <Hora-Salida> y llega a <Hora-Llegada> .
(INFORMACIÓN) HORA-LLEGADA-V TIPO-TREN	el tren de vuelta es un <Tipo-Tren> y llega a <Hora-Llegada> .
(INFORMACIÓN) PRECIO TIPO-TREN	el billete en <Tipo-Tren> cuesta <Precio> euros .
(INFORMACIÓN) TIEMPO-RECORRIDO	la duración del viaje es de <Tiempo-Recorrido> .

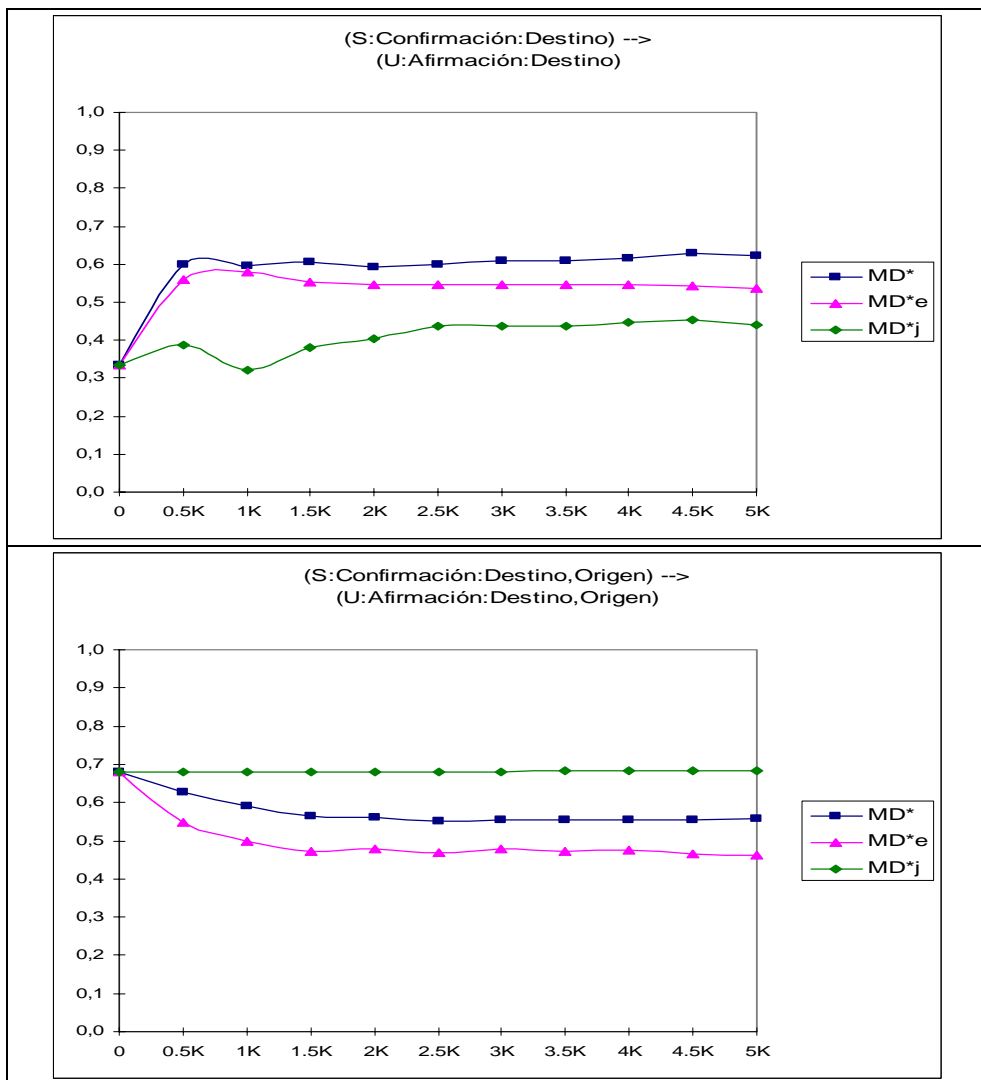
10.2 Simulador de usuario

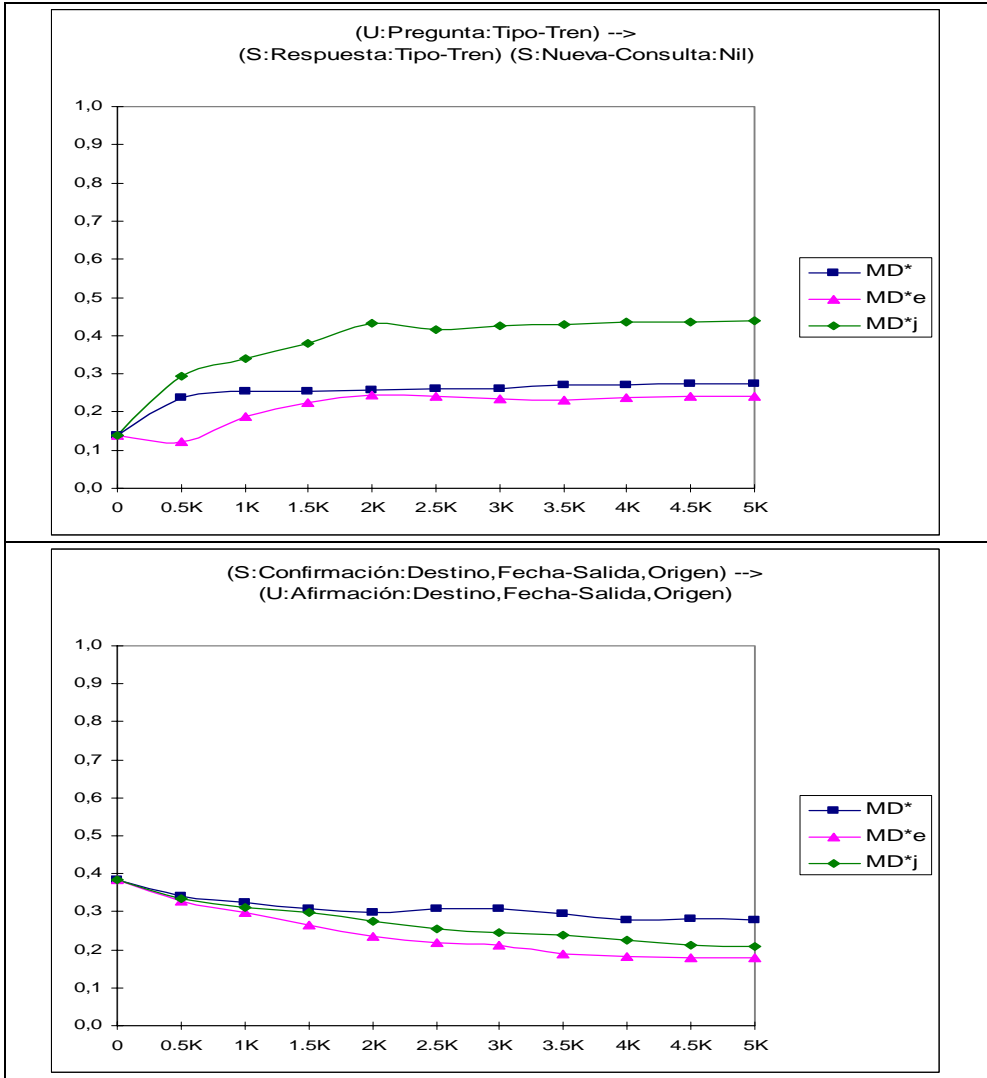
	ORIGEN	DESTINO	TIPO_TREN	PRECIO	FECHA SALIDA	HORA SALIDA	HORA LLEGADA	INTERVALO HORA SALIDA	INTERVALO HORA LLEGADA
1	Vigo	Soria	alaris	???	24-09-????	???	---	13.00-21.00	---
2	Toledo	Salamanca	---	---	04-11-????	???	???	05.00-13.00	---
3	Bilbao	Sevilla	---	???	24-12-????	???	---	13.00-21.00	---
4	Zamora	Granada	estrella	???	22-03-????	???	???	09.00-12.00	17.00-21.00
5	Valencia	Zaragoza	???	---	30-06-????	???	???	05.00-10.00	10.00-13.00
6	Albacete	Madrid	---	???	08-07-????	???	???	04.00-09.00	07.00-10.00
7	Valencia	Lérida	---	---	28-09-????	???	???	---	---
8	Lugo	Valencia	---	???	20-03-????	???	???	---	---
9	Barcelona	Bilbao	talgo	---	22-11-????	???	???	---	---
10	Valencia	Sevilla	---	???	31-07-????	---	---	---	---
11	Valencia	Alicante	---	---	06-12-????	???	???	---	---
12	Valencia	Madrid	talgo	???	11-11-????	???	???	04.00-10.00	07.00-12.00
13	Alicante	Salou	---	---	21-09-????	???	???	10.00-13.00	16.00-20.00
14	Monzón	Lérida	---	???	15-10-????	???	???	07.00-10.00	10.00-13.00

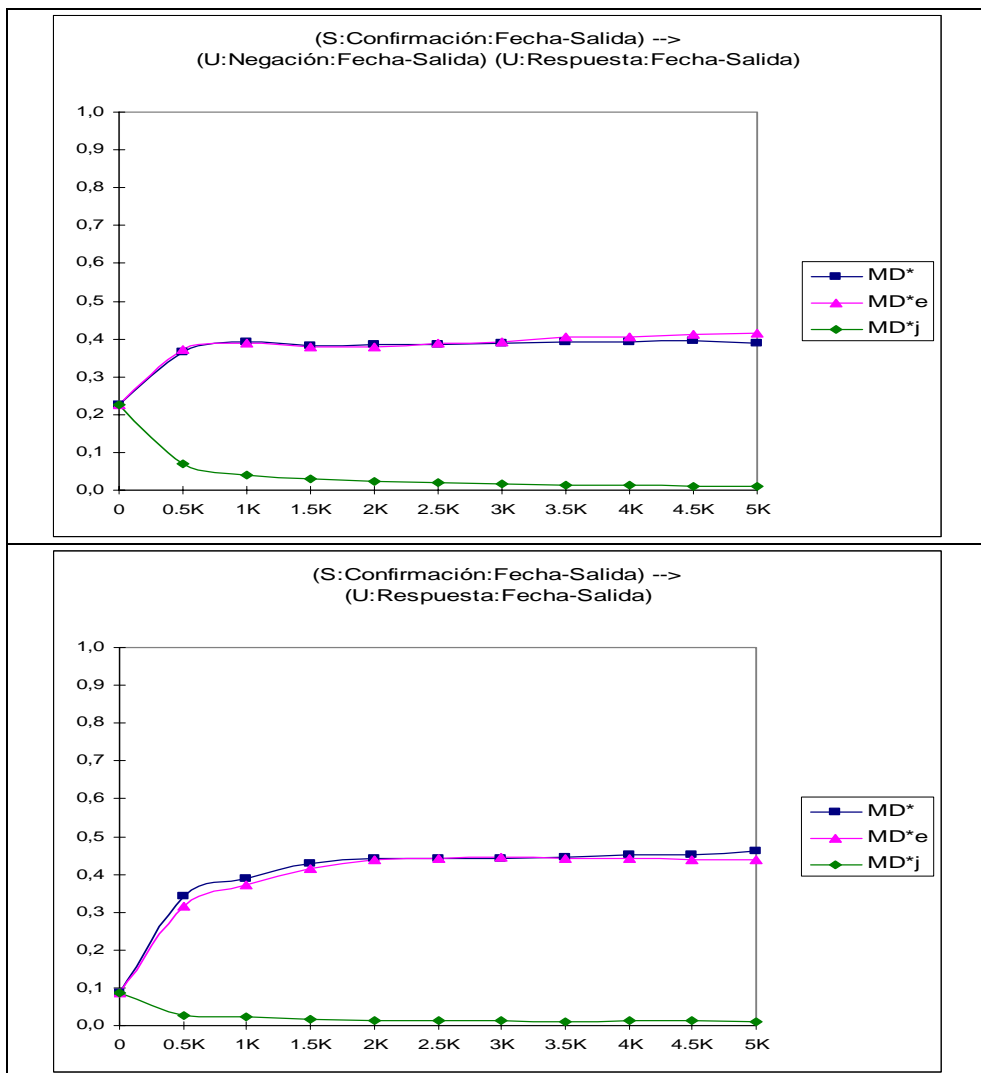
Tabla 10.3. Relación de escenarios usados en la simulación de diálogos

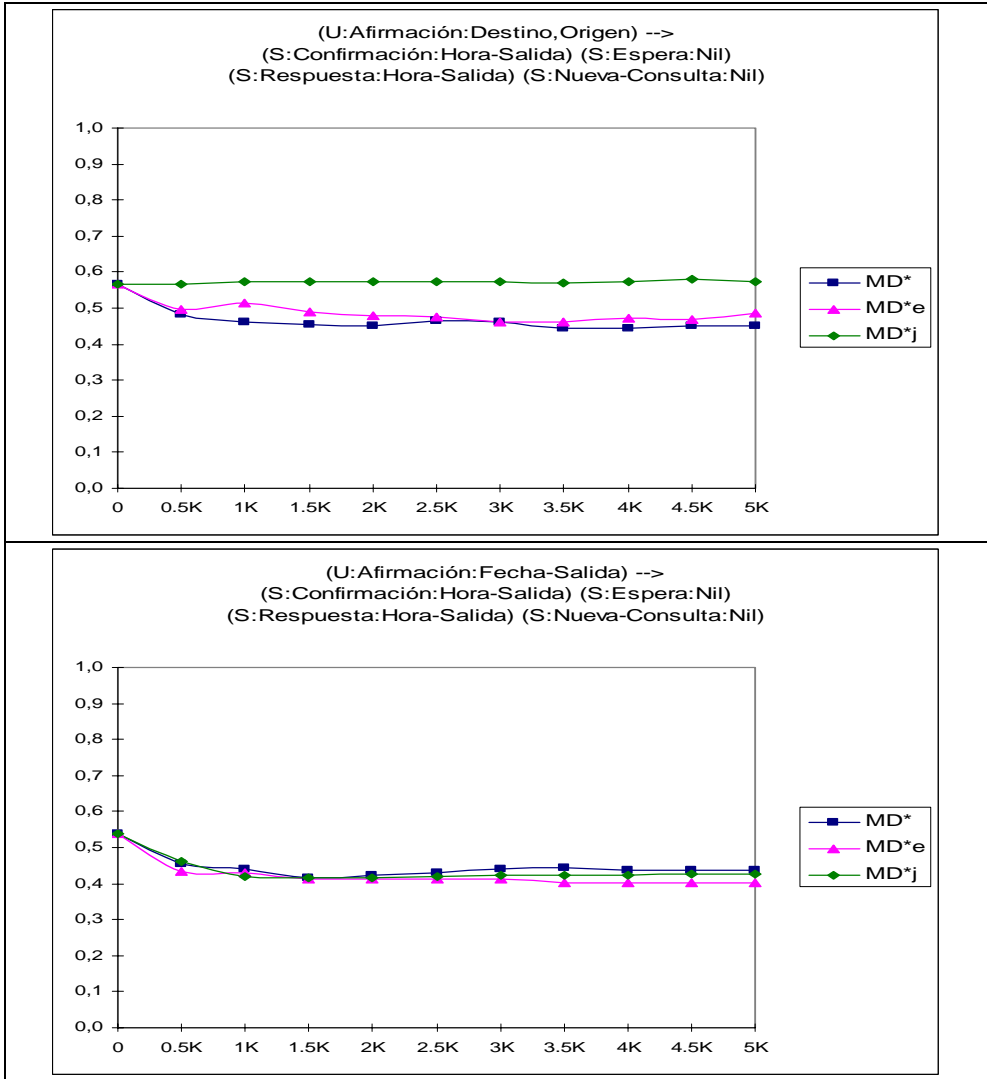












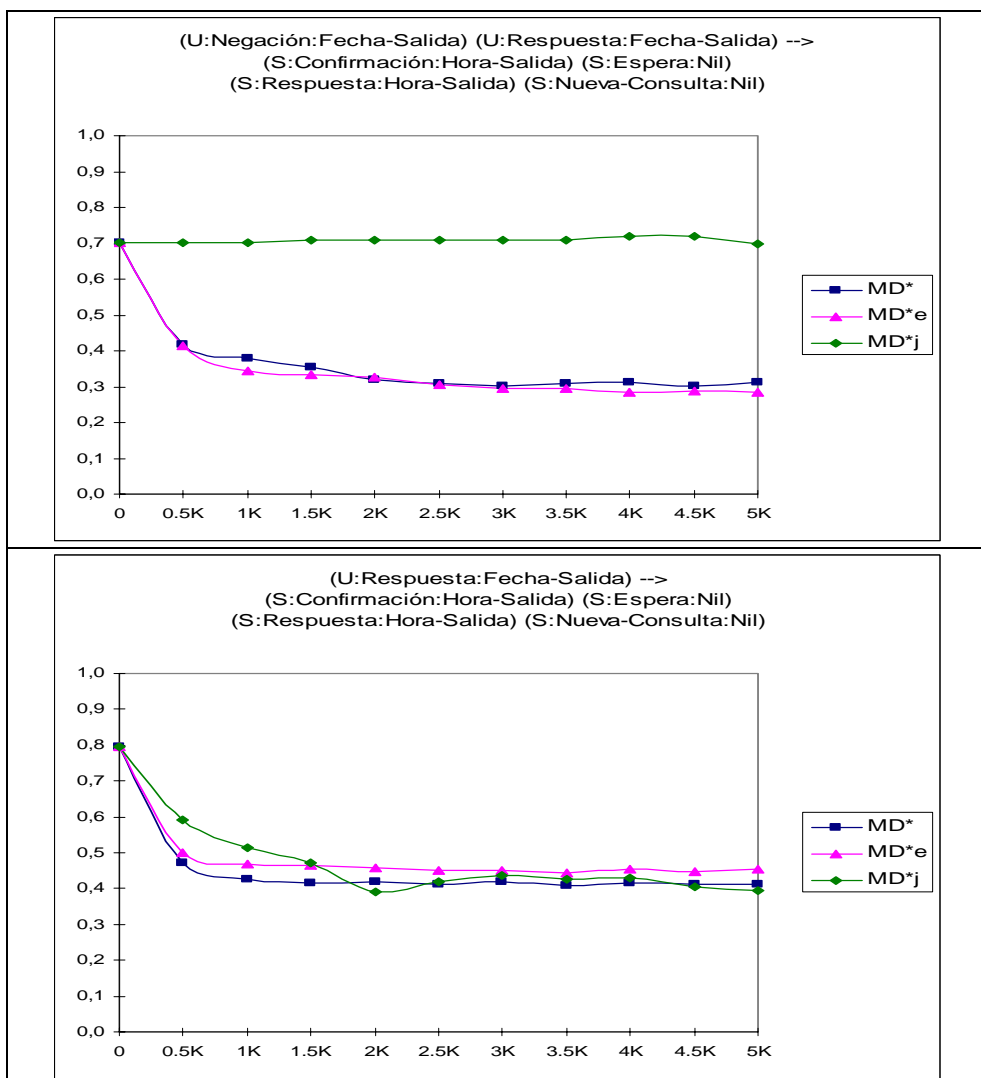


Figura 10.1: Evolución de las probabilidades de algunas transiciones durante el entrenamiento de los modelos MD*, MD*^e y MD*^j

Evaluación del modelo MD* ^c	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	258	51,60%	
Diálogos con introducción de errores	242	48,40%	
Diálogos concluidos con éxito	396	79,20%	
Diálogos con errores concluidos con éxito	138	57,02%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	411		0,82
Errores en atributo ORIGEN	130		0,26
Errores en atributo DESTINO	154		0,31
Errores en atributo FECHA-SALIDA	127		0,25
Turnos	2.406		4,81
Turnos (sistema) de Respuesta	1.189		2,38
Turnos (sistema) de Confirmación y Pregunta	42		0,08
Turnos (sistema) de Confirmación	812		1,62
Turnos (sistema) de Pregunta	358		0,72
Turnos (sistema) de Confirmación y Espera	0		0
Turnos (sistema) de sólo Nueva-Consulta	4		0,01
Turnos (sistema) de sólo No-Entendido	0		0
Transiciones ordinarias a estados usuario	2.405	99,96%	
Transiciones extraordinarias a estados usuario	1	0,04%	
Transiciones ordinarias a estados sistema	2.405	99,96%	
Transiciones extraordinarias a estados sistema	1	0,04%	

Tabla 10.4a: Evaluación del modelo MD*^c

Evaluación del modelo MD* ^c	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	252	50,40%	
Diálogos con introducción de errores	248	49,60%	
Diálogos concluidos con éxito	371	74,20%	
Diálogos con errores concluidos con éxito	119	47,98%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	425		0,85
Errores en atributo ORIGEN	127		0,25
Errores en atributo DESTINO	163		0,33
Errores en atributo FECHA-SALIDA	135		0,27
Turnos	2.342		4,68
Turnos (sistema) de Respuesta	1.189		2,38
Turnos (sistema) de Confirmación y Pregunta	36		0,07
Turnos (sistema) de Confirmación	794		1,59
Turnos (sistema) de Pregunta	318		0,64
Turnos (sistema) de Confirmación y Espera	0		0
Turnos (sistema) de sólo Nueva-Consulta	5		0,01
Turnos (sistema) de sólo No-Entendido	0		0
Transiciones ordinarias a estados usuario	2.341	99,96%	
Transiciones extraordinarias a estados usuario	1	0,04%	
Transiciones ordinarias a estados sistema	2.342	100%	
Transiciones extraordinarias a estados sistema	0	0	

Tabla 10.4b: Evaluación del modelo MD*^c

Evaluación del modelo MD* ^e	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	244	48,80%	
Diálogos con introducción de errores	256	51,20%	
Diálogos concluidos con éxito	375	75,00%	
Diálogos con errores concluidos con éxito	131	51,17%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	419		0,84
Errores en atributo ORIGEN	121		0,24
Errores en atributo DESTINO	165		0,33
Errores en atributo FECHA-SALIDA	133		0,27
Turnos	2.433		4,87
Turnos (sistema) de Respuesta	1.191		2,38
Turnos (sistema) de Confirmación y Pregunta	47		0,09
Turnos (sistema) de Confirmación	834		1,67
Turnos (sistema) de Pregunta	354		0,71
Turnos (sistema) de Confirmación y Espera	1		0
Turnos (sistema) de sólo Nueva-Consulta	6		0,01
Turnos (sistema) de sólo No-Entendido	0		0
Transiciones ordinarias a estados usuario	2.428	99,79%	
Transiciones extraordinarias a estados usuario	5	0,21%	
Transiciones ordinarias a estados sistema	2.433	100%	
Transiciones extraordinarias a estados sistema	0	0	

Tabla 10.4c: Evaluación del modelo MD*^e

Evaluación del modelo MD* ^e	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	231	46,20%	
Diálogos con introducción de errores	269	53,80%	
Diálogos concluidos con éxito	372	74,40%	
Diálogos con errores concluidos con éxito	141	52,42%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	454		0,91
Errores en atributo ORIGEN	134		0,27
Errores en atributo DESTINO	181		0,36
Errores en atributo FECHA-SALIDA	139		0,28
Turnos	2.438		4,88
Turnos (sistema) de Respuesta	1.191		2,38
Turnos (sistema) de Confirmación y Pregunta	45		0,09
Turnos (sistema) de Confirmación	816		1,63
Turnos (sistema) de Pregunta	378		0,76
Turnos (sistema) de Confirmación y Espera	6		0,01
Turnos (sistema) de sólo Nueva-Consulta	2		0
Turnos (sistema) de sólo No-Entendido	0		0
Transiciones ordinarias a estados usuario	2.437	99,96%	
Transiciones extraordinarias a estados usuario	1	0,04%	
Transiciones ordinarias a estados sistema	2.438	100%	
Transiciones extraordinarias a estados sistema	0	0	

Tabla 10.4d: Evaluación del modelo MD*^e

Evaluación del modelo MD*	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	232	46,40%	
Diálogos con introducción de errores	268	53,60%	
Diálogos concluidos con éxito	360	72,00%	
Diálogos con errores concluidos con éxito	128	47,76%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	460		0,92
Errores en atributo ORIGEN	144		0,29
Errores en atributo DESTINO	175		0,35
Errores en atributo FECHA-SALIDA	141		0,28
Turnos	2.348		4,70
Turnos (sistema) de Respuesta	1.194		2,39
Turnos (sistema) de Confirmación y Pregunta	25		0,05
Turnos (sistema) de Confirmación	806		1,61
Turnos (sistema) de Pregunta	316		0,63
Turnos (sistema) de Confirmación y Espera	6		0,01
Turnos (sistema) de sólo Nueva-Consulta	1		0
Turnos (sistema) de sólo No-Entendido	0		0
Transiciones ordinarias a estados usuario	2.348	100%	
Transiciones extraordinarias a estados usuario	0	0	
Transiciones ordinarias a estados sistema	2.348	100%	
Transiciones extraordinarias a estados sistema	0	0	

Tabla 10.5a: Evaluación del modelo MD*

Evaluación del modelo MD*	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	240	48,00%	
Diálogos con introducción de errores	260	52,00%	
Diálogos concluidos con éxito	380	76,00%	
Diálogos con errores concluidos con éxito	140	53,85%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	430		0,86
Errores en atributo ORIGEN	135		0,27
Errores en atributo DESTINO	165		0,33
Errores en atributo FECHA-SALIDA	130		0,26
Turnos	2.411		4,82
Turnos (sistema) de Respuesta	1.190		2,38
Turnos (sistema) de Confirmación y Pregunta	29		0,06
Turnos (sistema) de Confirmación	826		1,65
Turnos (sistema) de Pregunta	352		0,70
Turnos (sistema) de Confirmación y Espera	8		0,02
Turnos (sistema) de sólo Nueva-Consulta	6		0,01
Turnos (sistema) de sólo No-Entendido	0		0
Transiciones ordinarias a estados usuario	2.408	99,88%	
Transiciones extraordinarias a estados usuario	3	0,12%	
Transiciones ordinarias a estados sistema	2.411	100%	
Transiciones extraordinarias a estados sistema	0	0	

Tabla 10.5b: Evaluación del modelo MD*

Evaluación del modelo MD*	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	253	50,60%	
Diálogos con introducción de errores	247	49,40%	
Diálogos concluidos con éxito	368	73,60%	
Diálogos con errores concluidos con éxito	115	46,56%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	429		0,86
Errores en atributo ORIGEN	138		0,28
Errores en atributo DESTINO	158		0,32
Errores en atributo FECHA-SALIDA	133		0,27
Turnos	2.434		4,87
Turnos (sistema) de Respuesta	1.179		2,36
Turnos (sistema) de Confirmación y Pregunta	49		0,10
Turnos (sistema) de Confirmación	843		1,69
Turnos (sistema) de Pregunta	350		0,70
Turnos (sistema) de Confirmación y Espera	8		0,02
Turnos (sistema) de sólo Nueva-Consulta	5		0,01
Turnos (sistema) de sólo No-Entendido	0		0
Transiciones ordinarias a estados usuario	2.429	99,79%	
Transiciones extraordinarias a estados usuario	5	0,21%	
Transiciones ordinarias a estados sistema	2.434	100%	
Transiciones extraordinarias a estados sistema	0	0	

Tabla 10.5c: Evaluación del modelo MD*

Evaluación del modelo MD*	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	254	50,80%	
Diálogos con introducción de errores	246	49,20%	
Diálogos concluidos con éxito	375	75,00%	
Diálogos con errores concluidos con éxito	121	49,19%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	423		0,85
Errores en atributo ORIGEN	136		0,27
Errores en atributo DESTINO	160		0,32
Errores en atributo FECHA-SALIDA	127		0,25
Turnos	2.369		4,74
Turnos (sistema) de Respuesta	1.189		2,38
Turnos (sistema) de Confirmación y Pregunta	35		0,07
Turnos (sistema) de Confirmación	792		1,58
Turnos (sistema) de Pregunta	342		0,68
Turnos (sistema) de Confirmación y Espera	6		0,01
Turnos (sistema) de sólo Nueva-Consulta	5		0,01
Turnos (sistema) de sólo No-Entendido	0		0
Transiciones ordinarias a estados usuario	2.365	99,83%	
Transiciones extraordinarias a estados usuario	4	0,17%	
Transiciones ordinarias a estados sistema	2.369	100%	
Transiciones extraordinarias a estados sistema	0	0	

Tabla 10.5d: Evaluación del modelo MD*

Evaluación del modelo MD	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	252	50,40%	
Diálogos con introducción de errores	248	49,60%	
Diálogos concluidos con éxito	388	77,60%	
Diálogos con errores concluidos con éxito	136	54,84%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	421		0,84
Errores en atributo ORIGEN	121		0,24
Errores en atributo DESTINO	170		0,34
Errores en atributo FECHA-SALIDA	130		0,26
Turnos	2.419		4,84
Turnos (sistema) de Respuesta	1.197		2,39
Turnos (sistema) de Confirmación y Pregunta	38		0,08
Turnos (sistema) de Confirmación	795		1,59
Turnos (sistema) de Pregunta	304		0,61
Turnos (sistema) de Confirmación y Espera	32		0,06
Turnos (sistema) de sólo Nueva-Consulta	25		0,05
Turnos (sistema) de sólo No-Entendido	9		0,02
Otros Turnos (sistema)	19		0,04
Transiciones ordinarias a estados usuario	2.415	99,83%	
Transiciones extraordinarias a estados usuario	4	0,17%	
Transiciones ordinarias a estados sistema	2.400	99,21%	
Transiciones extraordinarias a estados sistema	19	0,79%	

Tabla 10.6a: Evaluación del modelo MD

Evaluación del modelo MD	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	252	50,40%	
Diálogos con introducción de errores	248	49,60%	
Diálogos concluidos con éxito	373	74,60%	
Diálogos con errores concluidos con éxito	121	48,79%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	425		0,85
Errores en atributo ORIGEN	146		0,29
Errores en atributo DESTINO	158		0,32
Errores en atributo FECHA-SALIDA	121		0,24
Turnos	2.478		4,96
Turnos (sistema) de Respuesta	1.198		2,40
Turnos (sistema) de Confirmación y Pregunta	49		0,10
Turnos (sistema) de Confirmación	809		1,62
Turnos (sistema) de Pregunta	339		0,68
Turnos (sistema) de Confirmación y Espera	51		0,10
Turnos (sistema) de sólo Nueva-Consulta	9		0,02
Turnos (sistema) de sólo No-Entendido	10		0,02
Otros Turnos (sistema)	13		0,03
Transiciones ordinarias a estados usuario	2.472	99,76%	
Transiciones extraordinarias a estados usuario	6	0,24%	
Transiciones ordinarias a estados sistema	2.465	99,48%	
Transiciones extraordinarias a estados sistema	13	0,52%	

Tabla 10.6b: Evaluación del modelo MD

Evaluación del modelo MD	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	233	46,60%	
Diálogos con introducción de errores	267	53,40%	
Diálogos concluidos con éxito	381	76,20%	
Diálogos con errores concluidos con éxito	148	55,43%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	422		0,84
Errores en atributo ORIGEN	126		0,25
Errores en atributo DESTINO	167		0,33
Errores en atributo FECHA-SALIDA	129		0,26
Turnos	2.476		4,95
Turnos (sistema) de Respuesta	1.184		2,37
Turnos (sistema) de Confirmación y Pregunta	40		0,08
Turnos (sistema) de Confirmación	806		1,61
Turnos (sistema) de Pregunta	346		0,69
Turnos (sistema) de Confirmación y Espera	55		0,11
Turnos (sistema) de sólo Nueva-Consulta	17		0,03
Turnos (sistema) de sólo No-Entendido	6		0,01
Otros Turnos (sistema)	22		0,04
Transiciones ordinarias a estados usuario	2.471	99,80%	
Transiciones extraordinarias a estados usuario	5	0,20%	
Transiciones ordinarias a estados sistema	2.454	99,11%	
Transiciones extraordinarias a estados sistema	22	0,89%	

Tabla 10.6c: Evaluación del modelo MD

Evaluación del modelo MD	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	254	50,80%	
Diálogos con introducción de errores	246	49,20%	
Diálogos concluidos con éxito	373	74,60%	
Diálogos con errores concluidos con éxito	119	48,37%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	426		0,85
Errores en atributo ORIGEN	136		0,27
Errores en atributo DESTINO	157		0,31
Errores en atributo FECHA-SALIDA	133		0,27
Turnos	2.424		4,85
Turnos (sistema) de Respuesta	1.185		2,37
Turnos (sistema) de Confirmación y Pregunta	48		0,10
Turnos (sistema) de Confirmación	783		1,57
Turnos (sistema) de Pregunta	333		0,67
Turnos (sistema) de Confirmación y Espera	32		0,06
Turnos (sistema) de sólo Nueva-Consulta	17		0,03
Turnos (sistema) de sólo No-Entendido	2		0
Otros Turnos (sistema)	24		0,05
Transiciones ordinarias a estados usuario	2.418	99,75%	
Transiciones extraordinarias a estados usuario	6	0,25%	
Transiciones ordinarias a estados sistema	2.400	99,01%	
Transiciones extraordinarias a estados sistema	24	0,99%	

Tabla 10.6d: Evaluación del modelo MD

Evaluación del modelo MD* ^e	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	48	9,60%	
Diálogos con introducción de errores	452	90,40%	
Diálogos concluidos con éxito	128	25,60%	
Diálogos con errores concluidos con éxito	80	16,00%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	824		1,65
Errores en atributo ORIGEN	264		0,53
Errores en atributo DESTINO	284		0,57
Errores en atributo FECHA-SALIDA	276		0,55
Turnos	1.947		3,89
Turnos (sistema) de Respuesta	1.178		2,36
Turnos (sistema) de Confirmación y Pregunta	21		0,04
Turnos (sistema) de Confirmación	439		0,88
Turnos (sistema) de Pregunta	299		0,60
Turnos (sistema) de Confirmación y Espera	1		0
Turnos (sistema) de sólo Nueva-Consulta	7		0,01
Turnos (sistema) de sólo No-Entendido	0		0
Otros Turnos (sistema)	2		0
Transiciones ordinarias a estados usuario	1.919	98,56%	
Transiciones extraordinarias a estados usuario	28	1,44%	
Transiciones ordinarias a estados sistema	1.945	99,90%	
Transiciones extraordinarias a estados sistema	2	0,10%	

Tabla 10.7a: Evaluación del modelo MD*^e

Evaluación del modelo MD* ^e	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	62	12,40%	
Diálogos con introducción de errores	438	87,60%	
Diálogos concluidos con éxito	153	30,60%	
Diálogos con errores concluidos con éxito	91	20,78%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	774		1,55
Errores en atributo ORIGEN	251		0,50
Errores en atributo DESTINO	294		0,59
Errores en atributo FECHA-SALIDA	229		0,46
Turnos	1.999		4,00
Turnos (sistema) de Respuesta	1.185		2,37
Turnos (sistema) de Confirmación y Pregunta	23		0,05
Turnos (sistema) de Confirmación	450		0,90
Turnos (sistema) de Pregunta	320		0,64
Turnos (sistema) de Confirmación y Espera	5		0,01
Turnos (sistema) de sólo Nueva-Consulta	3		0,01
Turnos (sistema) de sólo No-Entendido	0		0
Otros Turnos (sistema)	13		0,03
Transiciones ordinarias a estados usuario	1.971	98,60%	
Transiciones extraordinarias a estados usuario	28	1,40%	
Transiciones ordinarias a estados sistema	1.986	99,35%	
Transiciones extraordinarias a estados sistema	13	0,65%	

Tabla 10.7b: Evaluación del modelo MD*^e

Evaluación del modelo MD	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	61	12,20%	
Diálogos con introducción de errores	439	87,80%	
Diálogos concluidos con éxito	148	29,60%	
Diálogos con errores concluidos con éxito	87	19,82%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	834		1,67
Errores en atributo ORIGEN	266		0,53
Errores en atributo DESTINO	295		0,59
Errores en atributo FECHA-SALIDA	273		0,55
Turnos	2.035		4,07
Turnos (sistema) de Respuesta	1.176		2,35
Turnos (sistema) de Confirmación y Pregunta	27		0,05
Turnos (sistema) de Confirmación	428		0,86
Turnos (sistema) de Pregunta	303		0,61
Turnos (sistema) de Confirmación y Espera	26		0,05
Turnos (sistema) de sólo Nueva-Consulta	24		0,05
Turnos (sistema) de sólo No-Entendido	9		0,02
Otros Turnos (sistema)	42		0,08
Transiciones ordinarias a estados usuario	1.993	97,94%	
Transiciones extraordinarias a estados usuario	42	2,06%	
Transiciones ordinarias a estados sistema	1.993	97,94%	
Transiciones extraordinarias a estados sistema	42	2,06%	

Tabla 10.8a: Evaluación del modelo MD

Evaluación del modelo MD	Nº total	%	Nº medio por diálogo
Diálogos	500		
Diálogos sin introducción de errores	58	11,60%	
Diálogos con introducción de errores	442	88,40%	
Diálogos concluidos con éxito	158	31,60%	
Diálogos con errores concluidos con éxito	100	22,62%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	795		1,59
Errores en atributo ORIGEN	264		0,53
Errores en atributo DESTINO	267		0,53
Errores en atributo FECHA-SALIDA	264		0,53
Turnos	2.059		4,12
Turnos (sistema) de Respuesta	1.177		2,35
Turnos (sistema) de Confirmación y Pregunta	17		0,03
Turnos (sistema) de Confirmación	462		0,92
Turnos (sistema) de Pregunta	307		0,61
Turnos (sistema) de Confirmación y Espera	22		0,04
Turnos (sistema) de sólo Nueva-Consulta	18		0,04
Turnos (sistema) de sólo No-Entendido	9		0,02
Otros Turnos (sistema)	47		0,09
Transiciones ordinarias a estados usuario	2.008	97,52%	
Transiciones extraordinarias a estados usuario	51	2,48%	
Transiciones ordinarias a estados sistema	2.012	97,72%	
Transiciones extraordinarias a estados sistema	47	2,28%	

Tabla 10.8b: Evaluación del modelo MD

Evaluación del modelo MD	Nº total	%	Nº medio por diálogo
Diálogos	1.000		
Diálogos sin introducción de errores	436	43,60%	
Diálogos con introducción de errores	564	56,40%	
Diálogos concluidos con éxito	923	92,30%	
Diálogos con errores concluidos con éxito	487	86,35%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	1.173		1,17
Errores en atributo ORIGEN	391		0,39
Errores en atributo DESTINO	435		0,43
Errores en atributo FECHA-SALIDA	347		0,35
Turnos	4.539		4,54
Turnos (sistema) de Respuesta	2.400		2,40
Turnos (sistema) de Confirmación y Pregunta	59		0,06
Turnos (sistema) de Confirmación	1.401		1,40
Turnos (sistema) de Pregunta	546		0,55
Turnos (sistema) de Confirmación y Espera	39		0,04
Turnos (sistema) de sólo Nueva-Consulta	37		0,04
Turnos (sistema) de sólo No-Entendido	21		0,02
Otros Turnos (sistema)	36		0,04
Transiciones ordinarias a estados usuario	4.528	99,76%	
Transiciones extraordinarias a estados usuario	11	0,24%	
Transiciones ordinarias a estados sistema	4.503	99,21%	
Transiciones extraordinarias a estados sistema	36	0,79%	

Tabla 10.9: Evaluación del modelo MD

Evaluación del modelo MD* ^e	Nº total	%	Nº medio por diálogo
Diálogos	1.000		
Diálogos sin introducción de errores	481	48,10%	
Diálogos con introducción de errores	519	51,90%	
Diálogos concluidos con éxito	954	95,40%	
Diálogos con errores concluidos con éxito	473	91,14%	
Errores en ORIGEN, DESTINO o FECHA-SALIDA	1.047		1,05
Errores en atributo ORIGEN	372		0,37
Errores en atributo DESTINO	386		0,39
Errores en atributo FECHA-SALIDA	289		0,29
Turnos	4.427		4,43
Turnos (sistema) de Respuesta	2.414		2,41
Turnos (sistema) de Confirmación y Pregunta	46		0,05
Turnos (sistema) de Confirmación	1.354		1,35
Turnos (sistema) de Pregunta	606		0,61
Turnos (sistema) de Confirmación y Espera	1		0
Turnos (sistema) de sólo Nueva-Consulta	5		0,01
Turnos (sistema) de sólo No-Entendido	1		0
Transiciones ordinarias a estados usuario	4.421	99,86%	
Transiciones extraordinarias a estados usuario	6	0,14%	
Transiciones ordinarias a estados sistema	4.427	100%	
Transiciones extraordinarias a estados sistema	0	0	

Tabla 10.10: Evaluación del modelo MD*^e

10.3 Generador de respuestas del simulador de usuario

En la siguiente Tabla 10.11 se muestran los frames de usuario, generados por el gestor de diálogo de usuario (componente del simulador del usuario), en cada uno de los estados de usuario existentes en el modelo de diálogo.

	<i>Estado del modelo de diálogo</i>	<i>Frames de usuario</i>
102	(U:Afirmacion:Destino)	(AFIRMACION) 1.00
295	(U:Afirmacion:Destino,Dia_salida)	
22	(U:Afirmacion:Destino,Dia_salida,Origen)	
14	(U:Afirmacion:Destino,Origen)	
5	(U:Afirmacion:Dia_salida)	
131	(U:Afirmacion:Dia_salida,Origen)	
222	(U:Afirmacion:Hora_llegada)	
67	(U:Afirmacion:Hora_salida)	
89	(U:Afirmacion:Hora_salida_v)	
31	(U:Afirmacion:Origen)	
137	(U:Afirmacion:Servicio)	
125	(U:Afirmacion:Tipo_tren)	
50	(U:Afirmacion:Destino,Origen) (U:Respuesta:Dia_salida)	(AFIRMACION) 1.00 (INFORMACION) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd
255	(U:Afirmacion:Destino,Origen) (U:Respuesta:Dia_salida,Hora_salida)	(AFIRMACION) 1.00 (INFORMACION) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd HORA-SALIDA:dd.dd d.dd
85	(U:Afirmacion:Destino,Origen) (U:Respuesta:Hora_salida)	(AFIRMACION) 1.00 (INFORMACION) 1.00 HORA-SALIDA:dd.dd d.dd
134	(U:Afirmacion:Dia_salida) (U:Respuesta:Dia_salida,Hora_salida)	(AFIRMACION) 1.00 (INFORMACION) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd HORA-SALIDA:dd.dd d.dd
217	(U:Afirmacion:Dia_salida) (U:Respuesta:Origen)	(AFIRMACION) 1.00 (INFORMACION) 1.00 ORIGEN:ccc d.dd
216	(U:Afirmacion:Origen) (U:Respuesta:Hora_salida)	(AFIRMACION) 1.00 (INFORMACION) 1.00 HORA-SALIDA:dd.dd d.dd
56	(U:Apertura:Nil) (U:Pregunta:Hora_salida)	(APERTURA) 1.00 (SOLICITUD) 1.00 HORA-SALIDA:??? d.dd
73	(U:Apertura:Nil) (U:Pregunta:Hora_salida,Hora_salida_v)	(APERTURA) 1.00 (SOLICITUD) 1.00 HORA-SALIDA:??? d.dd HORA-SALIDA-V:??? d.dd

208	(U:Apertura:Nil) (U:Pregunta:Hora_salida,Hora_salida_v,Precio)	(APERTURA) 1.00 (SOLICITUD) 1.00 HORA-SALIDA:??? d.dd HORA-SALIDA-V:??? d.dd PRECIO:??? d.dd
70	(U:Apertura:Nil) (U:Pregunta:Hora_salida,Precio)	(APERTURA) 1.00 (SOLICITUD) 1.00 HORA-SALIDA:??? d.dd PRECIO:??? d.dd
11	(U:Cierre:Nil)	(CIERRE) 1.00
159	(U:Confirmacion:Dia_salida)	(CONFIRMACION) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd
99	(U:Confirmacion:Precio)	(CONFIRMACION) 1.00 PRECIO:dd d.dd
65	(U:Indefinida:Nil)	(INDEFINIDA) 1.00
186	(U:Negacion:Destino,Dia_salida,Origen)	(NEGACION) 1.00
177	(U:Negacion:Destino,Origen)	
107	(U:Negacion:Dia_salida)	
280	(U:Negacion:Tipo_tren)	
139	(U:Negacion:Dia_salida) (U:Pregunta:Hora_salida)	(NEGACION) 1.00 (SOLICITUD) 1.00
252	(U:Negacion:Origen) (U:Pregunta:Hora_salida)	HORA-SALIDA:??? d.dd
210	(U:Negacion:Destino) (U:Respuesta:Destino)	(NEGACION) 1.00 (INFORMACION) 1.00 DESTINO:ccc d.dd
233	(U:Negacion:Destino,Dia_salida,Origen) (U:Respuesta:Destino,Dia_salida,Origen)	(NEGACION) 1.00 (INFORMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd FECHA-SALIDA:dd-dd-dddd d.dd
142	(U:Negacion:Destino,Dia_salida,Origen) (U:Respuesta:Dia_salida)	(NEGACION) 1.00 (INFORMACION) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd
172	(U:Negacion:Destino,Origen) (U:Respuesta:Destino,Dia_salida,Origen)	(NEGACION) 1.00 (INFORMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd FECHA-SALIDA:dd-dd-dddd d.dd
84	(U:Negacion:Destino,Origen) (U:Respuesta:Destino,Origen)	(NEGACION) 1.00 (INFORMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd
39	(U:Negacion:Dia_salida) (U:Respuesta:Dia_salida)	(NEGACION) 1.00 (INFORMACION) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd
158	(U:Negacion:Dia_salida) (U:Respuesta:Dia_salida,Hora_salida)	(NEGACION) 1.00 (INFORMACION) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd HORA-SALIDA:dd.dd d.dd

237	(U:Negacion:Hora_salida) (U:Respuesta:Hora_salida)	(NEGACION) 1.00 (INFORMACION) 1.00 HORA-SALIDA:dd.dd d.dd
77	(U:No_entendido:Nil)	(NO-ENTENDIDO) 1.00
63	(U:Pregunta:Destino)	(SOLICITUD) 1.00 DESTINO:??? d.dd
163	(U:Pregunta:Dia_llegada)	(SOLICITUD) 1.00 FECHA-LLEGADA:??? d.dd
2	(U:Pregunta:Dia_salida)	(SOLICITUD) 1.00 FECHA-SALIDA:??? d.dd
26	(U:Pregunta:Hora_llegada)	(SOLICITUD) 1.00 HORA-LLEGADA:??? d.dd
34	(U:Pregunta:Hora_llegada,Hora_salida)	(SOLICITUD) 1.00 HORA-LLEGADA:??? d.dd HORA-SALIDA:??? d.dd
87	(U:Pregunta:Hora_llegada,Hora_salida,Precio)	(SOLICITUD) 1.00 HORA-LLEGADA:??? d.dd HORA-SALIDA:??? d.dd PRECIO:??? d.dd
261	(U:Pregunta:Hora_llegada,Hora_salida,Tipo_tren)	(SOLICITUD) 1.00 HORA-LLEGADA:??? d.dd HORA-SALIDA:??? d.dd TIPO-TREN: ??? d.dd
36	(U:Pregunta:Hora_llegada,Precio)	(SOLICITUD) 1.00 HORA-LLEGADA:??? d.dd PRECIO:??? d.dd
169	(U:Pregunta:Hora_llegada_v)	(SOLICITUD) 1.00 HORA-LLEGADA-V:??? d.dd
104	(U:Pregunta:Hora_llegada_v,Hora_salida_v)	(SOLICITUD) 1.00 HORA-LLEGADA-V:??? d.dd HORA-SALIDA-V:??? d.dd
231	(U:Pregunta:Hora_llegada_v,Precio,Tipo_tren)	(SOLICITUD) 1.00 HORA-LLEGADA-V:??? d.dd PRECIO:??? d.dd TIPO-TREN: ??? d.dd
213	(U:Pregunta:Hora_llegada_v,Tipo_tren)	(SOLICITUD) 1.00 HORA-LLEGADA-V:??? d.dd TIPO-TREN: ??? d.dd
8	(U:Pregunta:Hora_salida)	(SOLICITUD) 1.00 HORA-SALIDA:??? d.dd
40	(U:Pregunta:Hora_salida,Hora_salida_v)	(SOLICITUD) 1.00 HORA-SALIDA:??? d.dd HORA-SALIDA-V:??? d.dd
38	(U:Pregunta:Hora_salida,Hora_salida_v,Precio)	(SOLICITUD) 1.00 HORA-SALIDA:??? d.dd HORA-SALIDA-V:??? d.dd PRECIO:??? d.dd
43	(U:Pregunta:Hora_salida,Precio)	(SOLICITUD) 1.00 HORA-SALIDA:??? d.dd PRECIO:??? d.dd

193	(U:Pregunta:Hora_salida,Precio,Tipo_tren)	(SOLICITUD) 1.00 HORA-SALIDA:??? d.dd PRECIO:??? d.dd TIPO-TREN: ??? d.dd
272	(U:Pregunta:Hora_salida,Servicio)	(SOLICITUD) 1.00 HORA-SALIDA:??? d.dd SERVICIO:??? D.dd
270	(U:Pregunta:Hora_salida,Servicio,Tipo_tren)	(SOLICITUD) 1.00 HORA-SALIDA:??? d.dd SERVICIO:??? d.dd TIPO-TREN: ??? d.dd
273	(U:Pregunta:Hora_salida,Tipo_tren)	(SOLICITUD) 1.00 HORA-SALIDA:??? d.dd TIPO-TREN: ??? d.dd
15	(U:Pregunta:Hora_salida_v)	(SOLICITUD) 1.00 HORA-SALIDA-V:??? d.dd
45	(U:Pregunta:Hora_salida_v,Precio)	(SOLICITUD) 1.00 HORA-SALIDA-V:??? d.dd PRECIO:??? d.dd
90	(U:Pregunta:Hora_salida_v,Precio,Tipo_tren)	(SOLICITUD) 1.00 HORA-SALIDA-V:??? d.dd PRECIO:??? d.dd TIPO-TREN: ??? d.dd
118	(U:Pregunta:Origen)	(SOLICITUD) 1.00 ORIGEN:??? d.dd
9	(U:Pregunta:Precio)	(SOLICITUD) 1.00 PRECIO:??? d.dd
266	(U:Pregunta:Precio,Tiempo_recorrido)	(SOLICITUD) 1.00 PRECIO:??? d.dd TIEMPO-RECORRIDO:??? d.dd
78	(U:Pregunta:Precio,Tipo_tren)	(SOLICITUD) 1.00 PRECIO:??? d.dd TIPO-TREN:??? d.dd
18	(U:Pregunta:Servicio)	(SOLICITUD) 1.00 SERVICIO:??? d.dd
113	(U:Pregunta:Tiempo_recorrido)	(SOLICITUD) 1.00 TIEMPO-RECORRIDO:??? d.dd
6	(U:Pregunta:Tipo_tren)	(SOLICITUD) 1.00 TIPO-TREN:??? d.dd
52	(U:Respuesta:Destino)	(INFORMACION) 1.00 DESTINO:ccc d.dd
42	(U:Respuesta:Destino,Dia_salida,Origen)	(INFORMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd FECHA-SALIDA:dd-dd-dddd d.dd
72	(U:Respuesta:Destino,Origen)	(INFORMACION) 1.00 ORIGEN:ccc d.dd DESTINO:ccc d.dd
55	(U:Respuesta:Dia_salida)	(INFORMACION) 1.00 FECHA-SALIDA:dd-dd-dddd d.dd
269	(U:Respuesta:Hora_llegada)	(INFORMACION) 1.00 HORA-LLEGADA:dd.dd d.dd

61	(U:Respuesta:Hora_salida)	(INFORMACION) 1.00 HORA-SALIDA:dd.dd d.dd
211	(U:Respuesta:Origen)	(INFORMACION) 1.00 ORIGEN:ccc d.dd
245	(U:Respuesta:Precio)	(INFORMACION) 1.00 PRECIO:dd d.dd
80	(U:Respuesta:Servicio)	(INFORMACION) 1.00 SERVICIO:ccc d.dd
24	(U:Respuesta:Tipo_tren)	(INFORMACION) 1.00 TIPO-TREN:ccc d.dd
190	(U:Respuesta:Tipo_tren) (U:Pregunta:Precio)	(INFORMACION) 1.00 TIPO-TREN:ccc d.dd (SOLICITUD) 1.00 PRECIO:??? d.dd

En la siguiente Tabla 10.12 se muestran las plantillas usadas para generar las frases de respuesta del simulador de usuario y sus correspondencias con los frames del simulador de usuario, recibidos desde el módulo gestor de diálogo de usuario. En la columna de las plantillas de frases, los nombres en cursiva y delimitados con los caracteres “<” y “>” representan nombres de variables atributo, que se sustituyen con la expresión, en lenguaje natural, de los correspondientes valores de atributos, cuando la plantilla se transforma en segmento de una frase de respuesta.

<i>Componentes de frames de usuario</i>	<i>Plantillas de frases</i>
(INFORMACIÓN) DESTINO	quiero ir a <Destino> .
(INFORMACIÓN) FECHA-SALIDA	quiero viajar el <Día-Salida> .
(INFORMACIÓN) HORA-LLEGADA	el que llega a <Hora-Llegada> .
(INFORMACIÓN) HORA-SALIDA	el que sale a <Hora-Salida> .
(INFORMACIÓN) ORIGEN	quiero salir desde <Origen> .
(INFORMACIÓN) PRECIO	el que cuesta <Precio> euros .
(INFORMACIÓN) SERVICIOS	quiero viajar en un tren con <Servicio>.
(INFORMACIÓN) TIPO-TREN	quiero viajar en <Tipo-Tren> .
(INFORMACIÓN) FECHA-SALIDA HORA-SALIDA	quiero viajar el <Día-Salida> en el que sale a <Hora-Salida> .
(INFORMACIÓN) ORIGEN DESTINO	quiero ir de <Origen> a <Destino> .
(INFORMACIÓN) ORIGEN DESTINO FECHA-SALIDA	quiero ir de <Origen> a <Destino> el <Día-Salida> .
(AFIRMACIÓN)	sí .
(AFIRMACIÓN) DESTINO	sí , a <Destino> .
(AFIRMACIÓN) HORA-LLEGADA	sí , <Hora-Llegada> .
(AFIRMACIÓN) HORA-SALIDA	sí , <Hora-Salida> .
(AFIRMACIÓN) HORA-SALIDA-V	sí , <Hora-Salida-V>
(AFIRMACIÓN) ORIGEN	sí , de <Origen> .
(AFIRMACIÓN) SERVICIOS	sí , con <Servicio>.
(AFIRMACIÓN) TIPO-TREN	sí , en <Tipo-Tren> .
(AFIRMACIÓN) FECHA-SALIDA	sí , el <Día-Salida> .
(AFIRMACIÓN) DESTINO FECHA-SALIDA	sí , a <Destino> el <Día-Salida> .
(AFIRMACIÓN) ORIGEN DESTINO	sí , de <Origen> a <Destino> .
(AFIRMACIÓN) ORIGEN FECHA-SALIDA	sí , de <Origen> el <Día-Salida> .
(AFIRMACIÓN) ORIGEN DESTINO FECHA-SALIDA	sí , de <Origen> a <Destino> el <Día-Salida> .
(NEGACIÓN)	no .
(NEGACIÓN) DESTINO	no , a <Destino>.
(NEGACIÓN) FECHA-SALIDA	no , el <Día-Salida> .
(NEGACIÓN) HORA-SALIDA	no , <Hora-Salida> .
(NEGACIÓN) ORIGEN	no , de <Origen>.
(NEGACIÓN) TIPO-TREN	no , en <Tipo-Tren> .
(NEGACIÓN) ORIGEN DESTINO	no , de <Origen> a <Destino>.
(NEGACIÓN) ORIGEN DESTINO FECHA-SALIDA	no , de <Origen> a <Destino> el <Día-Salida>.
(SOLICITUD) DESTINO	¿ a dónde va ?
(SOLICITUD) FECHA-LLEGADA	¿ qué día llega ?
(SOLICITUD) FECHA-SALIDA	¿ qué día sale ?
(SOLICITUD) HORA-LLEGADA	¿ a qué hora llega ?
(SOLICITUD) SERVICIOS	¿ qué servicios ofrece ?
(SOLICITUD) TIEMPO-RECORRIDO	¿ cuánto tarda ?

(SOLICITUD) TIPO-TREN	¿ qué tipo de tren es ?
(SOLICITUD) ORIGEN	¿ de dónde sale ?
(SOLICITUD) PRECIO	¿ cuánto cuesta ?
(SOLICITUD) HORA-LLEGADA-V	¿ a qué hora llega el de vuelta ?
(SOLICITUD) HORA-SALIDA-V	¿ a qué hora sale el de vuelta ?
(SOLICITUD) HORA-SALIDA	¿ a qué hora sale ?
(SOLICITUD) HORA-LLEGADA-V HORA-SALIDA-V	¿ a qué hora llega el de vuelta y a qué hora sale el de vuelta ?
(SOLICITUD) HORA-LLEGADA-V TIPO-TREN	¿ a qué hora llega el de vuelta y qué tipo de tren es ?
(SOLICITUD) HORA-SALIDA HORA-SALIDA-V	¿ a qué hora sale el de ida y el de vuelta ?
(SOLICITUD) HORA-SALIDA PRECIO	¿ a qué hora sale y cuánto cuesta ?
(SOLICITUD) HORA-SALIDA SERVICIOS	¿ a qué hora sale y qué servicios ofrece ?
(SOLICITUD) PRECIO TIEMPO-RECORRIDO	¿ cuánto cuesta y cuánto tarda ?
(SOLICITUD) PRECIO TIPO-TREN	¿ cuánto cuesta y qué tipo de tren es ?
(SOLICITUD) HORA-LLEGADA HORA-SALIDA	¿ a qué hora llega y a qué hora sale ?
(SOLICITUD) HORA-LLEGADA PRECIO	¿ a qué hora llega y cuánto cuesta ?
(SOLICITUD) HORA-SALIDA TIPO-TREN	¿ a qué hora sale y qué tipo de tren es ?
(SOLICITUD) HORA-SALIDA-V PRECIO	¿ a qué hora sale el de vuelta y cuánto cuesta ?
(SOLICITUD) HORA-SALIDA PRECIO TIPO-TREN	¿ a qué hora sale , cuánto cuesta y qué tipo de tren es ?
(SOLICITUD) HORA-SALIDA HORA-SALIDA-V PRECIO	¿ a qué hora sale el de ida y el de vuelta y cuánto cuesta ?
(SOLICITUD) HORA-SALIDA-V PRECIO TIPO-TREN	¿ a qué hora sale el de vuelta , cuánto cuesta y qué tipo de tren es ?
(SOLICITUD) HORA-LLEGADA-V PRECIO TIPO-TREN	¿ a qué hora llega el de vuelta , cuánto cuesta y qué tipo de tren es ?
(SOLICITUD) HORA-SALIDA SERVICIOS TIPO-TREN	¿ a qué hora sale , qué servicios ofrece y qué tipo de tren es ?
(SOLICITUD) HORA-LLEGADA HORA-SALIDA PRECIO	¿ a qué hora llega , a qué hora sale y cuánto cuesta ?
(SOLICITUD) HORA-LLEGADA HORA-SALIDA TIPO-TREN	¿ a qué hora llega , a qué hora sale y qué tipo de tren es ?
(CONFIRMACIÓN) FECHA-SALIDA	¿ me confirma el <Día-Salida> ?
(CONFIRMACIÓN) PRECIO	¿ me confirma el <Precio> ?
(INDEFINIDA)	pues no sabría ...
(NO-ENTENDIDO)	me puede repetir por favor .
(APERTURA)	hola .
(CIERRE)	nada más muchas gracias .

11. Referencias

11. Referencias

- [1] Aust, H., Oerder, M., 1994. *Database query generation from spoken sentences*. Proc. of 2nd Workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA). Kyoto, Japan. pp. 141–144.
- [2] Aust, H., Oerder, M., Seide, F., Steinbiss, V., 1995. *The Philips automatic train timetable information system*. Speech Communication, 17 (1995), 249–262.
- [3] Baggia, P., Kellner, A., Pérennou, G., Popovici, C., Sturm, J., Wessel, F., 1999. *Language modelling and spoken dialogue systems - the ARISE experience*. Proc. of 6th European Conference on Speech Communication and Technology (Eurospeech). Budapest, Hungary. vol. 4, pp. 1767–1770.
- [4] Baggia, P., Castagneri, G., Danieli, M., 2000. *Field trials of the Italian ARISE train timetable system*. Speech Communication, 31 (2000), 355–367.
- [5] Béchet, F., Gorin, A., Wright, J., Tür, D.H., 2004. *Detecting and extracting named entities from spontaneous speech in a mixed-initiative spoken dialogue context: How may I help you?* Speech Communication 42 (2004), 207–225.
- [6] Benedí, J.M., Varona, A., Lleida, E., 2004. *DIHANA: Sistema de diálogo para el acceso a la información mediante habla espontánea en diferentes entornos*. Actas de las III Jornadas en Tecnología del Habla, Valencia, España.
- [7] Bonafonte, A., Aibar, P., Castell, N., Lleida, E., Mariño, J., Sanchis, E., Torres, M., 2000. *Desarrollo de un sistema de diálogo oral en dominios restringidos*. Actas de las I Jornadas en Tecnología del Habla. Sevilla, España.
- [8] Bonneau-Maynard, H., Devillers, L., Rosset, S., 2000. *Predictive performance of dialog systems*. Proc. of 2nd International Conference on Language Resources and Evaluation (LREC). Athens, Greece.
- [9] Bouwman, G., Hulstijn, J., 1998. *Dialogue strategy redesign with reliability measures*. Proc. of 1st Internat. Conf. on LREC. Granada, España. pp. 191–198.
- [10] Bouwman, G., Sturm, J., Boves, L., 1999. *Incorporating confidence measures in the Dutch train timetable information system developed in the Arise project*. Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP). Phoenix, Arizona (USA). vol. 1, pp. 493–496.
- [11] Constantinides, P., Rudnicky, A., 1999. *Dialog analysis in the Carnegie Mellon Communicator*. Proc. of 6th Eurospeech. Budapest, Hungary. vol. 1, pp. 243–246.
- [12] Córdoba, R., San-Segundo, R., Montero, J.M., Colás, J., Ferreiros, J., Macías-Guarasa, J., Pardo, J.M., 2001. *An interactive directory assistance service for Spanish with large vocabulary recognition*. Proc. of 7th Eurospeech. Aalborg, Denmark. pp. 1279–1282.
- [13] CMU, 1999. *The CMU Communicator*.
<http://www.speech.cs.cmu.edu/Communicator/>.

- [14] Devillers, L., Bonneau-Maynard, H., 1998. *Evaluation of dialog strategies for a tourist information retrieval system*. Proc. of International Conference on Spoken Language Processing (ICSLP). Sidney, Australia. pp. 1187–1190.
- [15] Dybkjaer, L., Bernsen, N., Minker, W., 2004. *Evaluation and usability of multimodal spoken language dialogue systems*. Speech Communication 43 (2004), 33–54.
- [16] Eckert, W., Levin, E., Pieraccini, R., 1997. *User modeling for spoken dialogue system evaluation*. Proc. of Automatic Speech Recognition and Understanding (ASRU) – IEEE Workshop. Santa Barbara, California (USA).
- [17] Forney, J. G. D., 1973. *The Viterbi algorithm*. Proc. IEEE 61 (3), 268–278.
- [18] Fraser, N., Gilbert, G., 1991. *Simulating speech systems*. Computer Speech and Language 5, 81–99.
- [19] García, F., Hurtado, L., Sanchis, E., Segarra, E., 2003. *The incorporation of confidence measures to language understanding*. Proc. of the 6th Conference on Text Speech and Dialogue (TSD). LNAI 2807. Springer, pp. 165–172.
- [20] Glass, J., Flammia, G., Goodine, D., Phillips, M., Polifroni, J., Sakai, S., Seneff, S., Zue, V., 1995. *Multilingual spoken-language understanding in the MIT Voyager system*. Speech Communication 17 (1995), 1–18.
- [21] Glass, J., Weinstein, E., 2001. *SpeechBuilder: facilitating spoken dialogue system development*. Proc. of 7th Eurospeech. Aalborg, Denmark. vol. 1, pp. 1335–1338.
- [22] Griol, D., Torres, F., Hurtado, L.F., Sanchis, E., Segarra, E., 2005. *Different approaches to the dialogue management in the DIHANA project*. Proc. of the 10th Specom (International Conference on Speech and Computer). Patras, Greece. vol. 1, pp. 203–206.
- [23] Hacioglu, K., Ward, W., 2002. *A concept graph based confidence measure*. Proc. of ICASSP. Orlando, Florida (USA). vol. 1, pp. 225–228.
- [24] Hazen, T., Seneff, S., Polifroni, J., 2002. *Recognition confidence scoring and its use in speech understanding systems*. Computer Speech and Language 16, 49–67.
- [25] Hurtado, L.F., Segarra, E., García, F., Sanchis, E., 2004. *Language understanding using n-multigram models*. Proc. of 4th International Conference of Advances in Natural Language Processing, (EsTAL), Alicante, España, pp. 207–219.
- [26] Karsenty, L., 2002. *Shifting the design philosophy of spoken natural language dialogue: from invisible to transparent systems*. Kluwer ed., International Journal of Speech Technology 5, pp. 147–157.
- [27] Kellner, A., Rueber, B., Seide, F., Tran, B., 1997. *PADIS - An automatic telephone switchboard and directory information system*. Speech Communication 23 (1997), 95–111.
- [28] Lamel, L., Rosset, S., Gauvain, J., Bennacef, S., Garnier-Rizet, M., Prouts, B., 2000a. *The LIMSI ARISE system*. Speech Communication 31 (2000), 339–353.

- [29] Lamel, L., Rosset, S., Gauvain, J., 2000b. *Considerations in the design and evaluation of spoken language dialog systems*. Proc. of ICSLP. Beijing, China. vol. 4, pp. 5–8.
- [30] Levin, E., Pieraccini, R., 1995. *Concept-based spontaneous speech understanding system*. Proc. of 4th Eurospeech. Madrid, España. vol. 2, pp. 555–558.
- [31] Levin, E., Pieraccini, R., Eckert, W., 1998. *Using Markov decision process for learning dialogue strategies*. Proc. of ICASSP. Seattle, Washington (USA). vol. 1, pp. 201–204.
- [32] Levin, E., Pieraccini, R., Eckert, W., Di Fabbrizio, G., Narayanan, S., 1999. *Spoken language dialogue: from theory to practice*. Proc. of ASRU–IEEE Workshop. Keystone, Colorado (USA).
- [33] Levin, E., Pieraccini, R., Eckert, W., 2000a. *A stochastic model of human-machine interaction for learning dialog strategies*. IEEE Transactions on Speech and Audio Processing. vol. 8 (1), pp. 11–23.
- [34] Levin, E., Narayanan, S., Pieraccini, R., Biatov, K., Bocchieri, E., Di Fabbrizio, G., Eckert, W., Lee, S., Pokrovsky, A., Rahim, M., Ruscitti, P., Walker, M., 2000b. *The AT&T-Darpa Communicator mixed-initiative spoken dialog system*. Proc. of ICSLP. Beijing, China. pp. 122–125.
- [35] López-Cózar, R., Rubio, A. J., García, P., Díaz-Verdejo, J. E., López-Soler, J. M., 2000. *Sistema telefónico de información a viajeros*. Actas de las I Jornadas en Tecnología del Habla. Sevilla, España.
- [36] López-Cózar, R., Segura, J.C., De la Torre, A., Rubio, A.J., 2001. *Una nueva técnica para evaluar sistemas conversacionales basada en la generación automática de diálogos*. Procesamiento del Lenguaje Natural, vol. 27, pp. 255–260.
- [37] López-Cózar, R., De la Torre, A., Segura, J.C., Rubio, A.J., Sánchez, V., 2002. *Testing dialogue systems by means of automatic generation of conversations*. Interacting with Computers 14 (2002), pp. 521–546.
- [38] López-Cózar, R., De la Torre, A., Segura, J.C., Rubio, A.J., 2003. *Assessment of dialogue systems by means of a new simulation technique*. Speech Communication 40 (2003), 387–407.
- [39] Macherey, K., Bender, O., Ney, H., 2003a. *Multi-level error handling for tree based dialogue course management*. ISCA Tutorial and Research Workshop on Error Handling in Spoken Dialogue Systems. Chateau-d'Oex-Vaud, Switzerland. pp. 123–128.
- [40] Macherey, K., Ney, H., 2003b. *Features of tree based dialogue course management*. Proc. of 8th Eurospeech. Geneva, Switzerland. vol. 1, pp. 601–604.
- [41] Mariño, J., Hernando, J., 1999. *Especificación de las grabaciones mediante Mago de Oz*. Universidad Politécnica de Cataluña, Proyecto Basurde, Report de Investigación BS16AV10.

- [42] Martínez, C., Sanchis, E., García, F., Aibar, P., 2002. *A labeling proposal to annotate dialogues*. Proc. of 3rd Internat. Conf. on LREC. Las Palmas de Gran Canaria, España. pp. 1577–1582.
- [43] McGlashan, S., Burnett, D., Carter, J., Danielsen, P., Ferrans, J., Hunt, A., Lucas, B., Porter, B., Rehor, K., Tryphonas, S., 2004. *Voice Extensible Markup Language (VoiceXML) Version 2.0.* <http://www.w3.org/TR/voicexml20/>
- [44] Minker, W., Bennacef, S., Gauvain, J., 1996. A stochastic case frame approach for natural language understanding. Proc. of ICSLP. Philadelphia, Pennsylvania (USA). pp. 1013–1016.
- [45] Minker, W., 1997. *Stochastically-based natural language understanding across tasks and languages*. Proc. of 5th Eurospeech. Rhodes, Greece. vol. 3, pp. 1423–1426.
- [46] Oh, A., Rudnicky, A., 2000. *Stochastic language generation for spoken dialogue systems*. ANLP/NAACL 2000 Workshop on Conversational Systems, pp. 27–32.
- [47] Oh, A., Rudnicky, A., 2002. *Stochastic natural language generation for spoken dialog systems*. Computer Speech and Language, 16 (2002), 387–407.
- [48] Pargellis, A., Kuo, H.-K., Lee, C.-H., 2004. *An automatic dialogue generation platform for personalized dialogue applications*. Speech Communication 42 (2004), 329–351.
- [49] Peckham, J., 1993. *A new generation of spoken dialogue systems: results and lessons from the SUNDIAL project*. Proc. of 3rd Eurospeech. Berlin, Germany. vol. 1, pp. 33–42.
- [50] Pérez-Piñar, D., García, C., 2005. *Application of confidence measures for dialogue systems through the use of parallel speech recognizers*. Proc. of 9th Eurospeech. Lisboa, Portugal. pp. 2785–2788.
- [51] Pieraccini, R., Levin, E., Eckert, W., 1997. *AMICA: The AT&T mixed initiative conversational architecture*. Proc. of 5th Eurospeech. Rhodes, Greece. pp. 1875–1878.
- [52] Pietquin, O., 2004. *A framework for unsupervised learning of dialogue strategies*. Ph.D. thesis. Faculty of Engineering, Mons, Belgium.
- [53] Potamianos, A., Narayanan, S., Riccardi, G., 2005. *Adaptive categorical understanding for spoken dialogue systems*. IEEE Transactions on Speech and Audio Processing, vol. 13 (3), pp. 321–329.
- [54] Raymond, C., Estève, Y., Béchet, F., Mori, R. D., Damnati, G., 2003. *Belief confirmation in spoken dialog systems using confidence measures*. Proc. of 8th ASRU–IEEE Workshop. St. Thomas, U.S. Virgin Islands. pp. 150–155.
- [55] Raymond, C., Béchet, F., Mori, R. D., Damnati, G., Estève, Y., 2004. *Automatic learning of interpretation strategies for spoken dialogue systems*. Proc. of ICASSP. Montréal, Québec (Canada). vol. 1, pp. 425–428.

- [56] Riccardi, G., Gorin, A., 2000. *Stochastic language adaptation over time and state in natural spoken dialog systems*. IEEE Transactions on Speech and Audio Processing, vol. 8 (1), pp. 3–10.
- [57] Rosset, S., Bennacef, S., Lamel, L., 1999. *Design strategies for spoken language dialog systems*. Proc. of 6th Eurospeech. Budapest, Hungary. pp. 1535–1538.
- [58] Rudnicky, A., Xu W., 1999. *An agenda-based dialog management architecture for spoken language systems*. Proc. of ASRU–IEEE Workshop. Keystone, Colorado (USA). pp. I–337–340.
- [59] Rudnicky, A.I., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu W., Oh, A., 1999. *Creating natural dialogs in the Carnegie Mellon Communicator system*. Proc. of 6th Eurospeech. Budapest, Hungary. vol. 4, pp. 1531–1534.
- [60] San-Segundo, R., Montero, J. M., Gutiérrez, J. M., Gallardo, A., Romeral, J. D., Pardo, J.M., 2001a. *A telephone-based railway information system for spanish: development of a methodology for spoken dialogue design*. Proc. of the 2nd SIGdial Workshop on Discourse and Dialogue. Aalborg, Denmark. pp. 140–148.
- [61] San-Segundo, R., Pellom, B., Hacıoglu, K., Ward, W., Pardo, J.M., 2001b. *Confidence measures for spoken dialogue systems*. Proc. of ICASSP. Salt Lake City, Utah (USA). vol. 1, pp. 393–396.
- [62] Schatzmann, J., Georgila, K., Young, S., 2005. *Quantitative evaluation of user simulation techniques for spoken dialog systems*. Proc. of the 6th SIGdial Workshop on Discourse and Dialogue. Lisboa, Portugal. pp. xxx–xxx.
- [63] Scheffler, K., Young, S., 2001. *Corpus-based dialogue simulation for automatic strategy learning and evaluation*. Proc. of North American Chapter of the Association for Computational Linguistics (NAACL), Workshop on Adaptation in Dialogue Systems. Pittsburgh, Pennsylvania (USA). pp. 64–70.
- [64] Schwartz, R., Miller, S., Stallard, D., Makhoul, J., 1996. *Language understanding using hidden understanding models*. Proc. of ICSLP. Philadelphia, Pennsylvania (USA). vol. 2, pp. 997–1000.
- [65] Segarra, E., Arranz, V., Castell, N., Galiano, M., García, F., Molina, A., Sanchis, E., 2000. *Representación semántica de la tarea*. Informe técnico, proyecto BASURDE.
- [66] Segarra, E., Sanchis, E., Galiano, M., García, F., Hurtado, L., 2002. *Extracting semantic information through automatic learning techniques*. International Journal of Pattern Recognition and Artificial Intelligence 16 (3), 301–307.
- [67] Sheeder, T., Balogh, J., 2003. *Say it like you mean it: priming for structure in caller responses to a spoken dialog system*. Kluwer ed., International Journal of Speech Technology 6, pp. 103–111.
- [68] Souvignier, B., Kellner, A., Rueber, B., Schramm, H., Seide, F., 2000. *The thoughtful elephant: strategies for spoken dialog systems*. IEEE Transactions on Speech and Audio Processing. vol. 8 (1), pp. 51–62.

- [69] Stolcke A., Coccaro, N., Bates, R., Taylor, P., Ess-Dykema, C., Ries, K., Shriberg, E., Jurafsky, D., Martin, R., Meteer, M., 2000. *Dialogue act modeling for automatic tagging and recognition of conversational speech*. Association for Computational Linguistics (ACL), 2000. vol. 26 (3), pp. 339–373.
- [70] Sturm, J., den Os, E., Boves, L., 1999a. *Dialogue management in the Dutch Arise train timetable information system*. Proc. of 6th Eurospeech. Budapest, Hungary. vol. 3, pp. 1419–1422.
- [71] Sturm, J., den Os, E., Boves, L., 1999b. *Issues in spoken dialogue systems: experiences with the Dutch Arise system*. Proc. of ESCA Workshop on Interactive dialogue in multi-modal systems, Kloster Irsee, Germany. pp. 1–4.
- [72] Sturm, J., Wang, F., Cranen, B., 2001. *Adding extra input/output to a spoken dialogue system*. Proc. of 2nd SIGdial Workshop on Discourse and Dialogue. Aalborg, Denmark.
- [73] Sturm, J., Bakx, I., Cranen, B., Terken, J., Wang, F., 2002. *Usability evaluation of a Dutch multimodal system for train timetable information*. Proc. of 3rd Internat. Conf. on LREC. Las Palmas de Gran Canaria, España.
- [74] Torres, F., Sanchis, E., Segarra, E., 2002a. *Desarrollo de un gestor de diálogo basado en modelos estocásticos y dirigido por la semántica*. Procesamiento del Lenguaje Natural, vol. 29, pp. 175–180.
- [75] Torres, F., Sanchis, E., Segarra, E., 2002b. *El uso de medidas de confianza en un gestor de diálogo dirigido por la semántica*. Actas de las II Jornadas en Tecnología del Habla. Granada, España.
- [76] Torres, F., Sanchis, E., Segarra, E., 2003. *Development of a stochastic dialog manager driven by semantics*. Proc. of 8th Eurospeech. Geneva, Switzerland. vol. 1, pp. 605–608.
- [77] Torres, F., Sanchis, E., Segarra, E., 2004. *Un gestor de diálogo estocástico para el proyecto DIHANA*. Actas de las III Jornadas en Tecnología del Habla. Valencia, España. pp. 47–52.
- [78] Torres, F., Hurtado, L.F., García, F., Sanchis, E., Segarra, E., 2005a. *Error handling in a stochastic dialog system through confidence measures*. Special Issue on Error Handling in Spoken Dialogue Systems, Speech Communication 45 / 3, (2005), 211–229.
- [79] Torres, F., Sanchis, E., Segarra, E., 2005b. *Learning of stochastic dialog models through a dialog simulation technique*. Proc. of 9th Eurospeech. Lisboa, Portugal. pp. 817–820.
- [80] Walker M.A., Litman, D.J., Kamm, C.A., Abella, A., 1997. *Paradise: a general framework for evaluating spoken dialog agents*. Proc. of ACL/EACL, pp. 271–280.
- [81] Walker, M.A., Litman, D.J., Kamm, C.A., Abella, A., 1998. *Evaluating spoken dialogue agents with PARADISE: Two case studies*. Computer Speech and Language 12, 317–347.

- [82] Ward, W., Pellom, B., 1999. *The CU Communicator system*. Proc. of ASRU-IEEE Workshop. Keystone, Colorado (USA).
- [83] Wessel, F., Macherey, K., Schlüter, R., 1998. *Using word probabilities as confidence measures*. Proc. of ICASSP. Seattle, Washington (USA). vol. 1, pp. 225–228.
- [84] Wessel, F., Macherey, K., Ney, H., 1999. *A comparison of word graph and n-best list based confidence measures*. Proc. of 6th Eurospeech. Budapest, Hungary. pp. 315–318.
- [85] Williams, J.D., Young, S., 2003. *Using Wizard-of-Oz simulations to bootstrap reinforcement-learning-based dialog management systems*. Proc. of 4th SIGdial Workshop on Discourse and Dialogue. Sapporo, Japan.
- [86] Williams, J.D., Witt, S.M., 2004. *A comparison of dialog strategies for call routing*. Kluwer ed., International Journal of Speech Technology 7, 9–24.
- [87] Williams, J.D., Poupart, P., Young, S., 2005a. *Using factored partially observable Markov decision processes with continuous observations for dialog management*. Cambridge University, Dept. of Engineering, Technical Report CUED/F-INFENG/TR.520.
- [88] Williams, J.D., Poupart, P., Young, S., 2005b. *Partially observable Markov decision processes with continuous observations for dialog management*. Proc. of the 6th SIGdial Workshop on Discourse and Dialogue. Lisboa, Portugal. pp. xxx–xxx.
- [89] Young, S., 2002. *The statistical approach to the design of spoken dialogue systems*. Cambridge University, Dept. of Engineering, Technical Report CUED/F-INFENG/TR.433.
- [90] Zhang, R., Rudnicky, A., 2001. *Word level confidence annotation using combinations of features*. Proc. of 7th Eurospeech. Aalborg, Denmark. pp. 2105–2108.
- [91] Zue, V., Seneff, S., Polifroni, J., Phillips, M., Pao, C., Goodine, D., Goddeau, D., Glass, J., 1994. *PEGASUS: A spoken dialogue interface for on-line air travel planning*. Speech Communication 15 (1994), 331–340.
- [92] Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T., Hetherington, L., 2000. *JUPITER: A telephone-based conversational interface for weather information*. IEEE Transactions on Speech and Audio Processing. vol. 8 (1), pp. 85–96.