Procesamiento del Lenguaje Natural, num. 29 (2002), pp.175-180 recibido 09-05-2002; aceptado 02-07-2002

Desarrollo de un gestor de diálogo basado en modelos estocásticos y dirigido por la semántica

Francisco Torres, Emilio Sanchis, Encarna Segarra
Departamento de Sistemas Informáticos y Computación (DSIC)
Universidad Politécnica de Valencia (UPV)
Camino de Vera s/n, 46022 Valencia, Spain
{ftgoterr, esanchis, esegarra} @dsic.upv.es

Resumen: Presentamos una aproximación para el desarrollo de un gestor de diálogo basado en modelos estocásticos para representar la estructura y la estrategia de los diálogos. La entrada al gestor consiste en la representación semántica del turno de usuario. Esta aproximación se ha aplicado a un sistema de diálogo de acceso telefónico en castellano que contiene información sobre horarios de trenes.
Palabras clave: Diálogo, Gestor de Diálogo, Modelos Estocásticos, Actos de Diálogo.

Abstract: We present an approach to the development of a dialogue manager based on stochastic models for the representation of the dialogue structure and strategy. The input of the manager consists of the semantic representation of the user turn. It has been applied to a Spanish dialogue system which answers queries about train timetables by telephone in Spanish.
Keywords: Dialogue Systems, Dialogue Manager, Stochastic Models, Dialogue Acts.

1 Introducción

El acceso mediante diálogo hablado a sistemas de información es uno de los objetivos más interesantes dentro del área de las Tecnologías del Habla. Los avances en el análisis y modelización de las fuentes de conocimiento involucradas en las diferentes fases de un sistema de diálogo hablado, como son el reconocimiento del habla, la comprensión o la síntesis de habla, han permitido el desarrollo de prototipos que, aunque de posibilidades limitadas, abren una vía para posteriores progresos.

Algunas características de estos sistemas son: el acceso al sistema es telefónico, lo que les da una operatividad real; las tareas son de dominio restringido, ya que es preciso limitar su léxico y ámbito semántico para obtener sistemas viables; y finalmente, la iniciativa en el diálogo es mixta, para permitir que el usuario tenga cierta libertad en la generación de los turnos, resultando el diálogo más natural. La descripción de algunos de estos sistemas desarrollados en los últimos años se puede encontrar en Cmu-csdtk, (Pieraccini y Levin, 1997), (Lamel et al., 2000), (Glass y Weinstein, 01), (Córdoba, 2001) y (López-Cozar et al., 2002).

El trabajo que aquí se presenta es una aproximación al desarrollo del gestor de diálogo dentro de un sistema de diálogo. Se ha aplicado a un sistema de diálogo de acceso telefónico en castellano que contiene información sobre horarios de trenes, en el marco del proyecto BASURDE (Bonafonte et al., 2000). El gestor de diálogo propuesto está basado en la utilización de modelos estocásticos, estimados a partir de muestras, para representar la estructura y la estrategia de los diálogos (Martínez y Casacuberta, 2000). Sin embargo, este tipo de modelización, que tan buenos resultados da en campos, como el reconocimiento del habla, se encuentra con fuertes limitaciones cuando trata de representar las posibles situaciones de un diálogo, debido al escaso número de muestras de aprendizaje disponibles y a la gran variabilidad de situaciones (estados del diálogo).

Para abordar el problema del entrenamiento insuficiente por falta de muestras, se han utilizado dos modelos, uno más genérico y otro más específico, que se alternan durante el proceso de diálogo, así como una generalización de la representación semántica de entrada. Si, pese al aumento de la cobertura del modelo de diálogo que proporciona este proceso de generalización (esta especie de suavizado del modelo), el sistema se encuentra en situaciones no previstas, se arbitran procedimientos para que el proceso no se detenga o degenere en turnos sin sentido. Es decir, se arbitran reglas cuando el mecanismo puramente estocástico no puede proporcionar la respuesta esperada.

ISSN 1135-5948

2 La tarea BASURDE

La tarea definida en el proyecto consiste en consultas telefónicas sobre horarios, precios y servicios de trenes españoles de largo recorrido. A partir del análisis de un corpus de 200 diálogos persona–persona correspondientes a un sistema de información real, se definieron cuatro tipos de escenarios: horarios para viajes de ida, horarios para viajes de ida y vuelta, precios y servicios, y un escenario libre adicional. Se adquirieron 215 diálogos usando la técnica del Mago de Oz. El número total de turnos de usuario adquirido fue de 1.460 (14.902 palabras).

3 Etiquetado de los actos de diálogo

Los estados del diálogo se representan mediante actos de diálogo. Se definió un conjunto de actos de diálogo para la tarea y se les distinguió mediante etiquetas. Esta definición es muy importante pues determina el nivel de detalle en la representación del proceso del diálogo. Si se define un conjunto pequeño de etiquetas, sólo se podrá modelizar el propósito general del turno de diálogo. Si, por el contrario, se define un conjunto grande, cada etiqueta puede mostrar con más detalle la intención del turno, pero el modelo puede quedar pobremente estimado, dada la falta de muestras de aprendizaje.

En el proyecto BASURDE se ha propuesto un conjunto de etiquetas de actos de diálogo de tres niveles (Martínez et al., 2002). El primer nivel describe el comportamiento general del diálogo y es independiente de la tarea (*apertura*, *cierre*, *pregunta*, *confirmación*, …). El segundo nivel está relacionado con la representación semántica del turno y es específico de la tarea (*hora-salida*, *precio*, *tipo-tren*, …). En el tercer nivel se representan los valores (atributos) dados en el turno de diálogo. El modelo estocástico de gestión del diálogo se ha implementado a partir de los dos primeros niveles, quedando el tercer nivel implícito en la información semántica que recibe el gestor de diálogo en su entrada.

4 Representación semántica

El sistema de diálogo desarrollado en el proyecto BASURDE (Bonafonte et al., 2000) sigue un esquema modular. Se han definido los siguientes módulos: el reconocedor del habla, el módulo de comprensión, el gestor de diálogo y el módulo de generación de respuesta y síntesis. La entrada al gestor de diálogo es la representación semántica de los turnos del usuario. Esta información semántica, proporcionada por el módulo de comprensión del sistema (Segarra et al., 2001) produce el correspondiente cambio de estado del modelo, así como la actualización de los datos proporcionados hasta el momento. En este sistema, se utilizan *frames* para representar los turnos de usuario (Sanchis et al., 2000). En la Figura 1 se muestra un ejemplo de un turno de usuario y su correspondiente representación mediante *frames*.

```
U0: Hola buenas, quería viajar de Zaragoza a Bilbao 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?

U0: (HORA-SALIDA)
        CIUDAD-ORIGEN: Zaragoza
        CIUDAD-DESTINO: Bilbao
        FECHA-SALIDA: 02-11-2002
        INTERVALO-HORA-SALIDA: 05.00-13.00
    (HORA-SALIDA-V)
        FECHA-SALIDA: 05-11-2002
```

Figura 1: *Ejemplo de representación semántica*

5 Descripción del sistema

La interacción del proceso de diálogo se ajusta al algoritmo indicado en la Figura 2. La dirección del diálogo usuario–sistema es determinada por dos componentes: el modelo estocástico de diálogo (MD), y el registro de valores actuales (RVA). El diseño de estos componentes constituye, por tanto, la clave para alcanzar la operatividad del sistema gestor de diálogo.

```
Iniciar (RVA); /* RVA = Registro Valores Actuales */
Lectura (MD); /* MD = Autómatas Modelo Diálogo */
estado_MD = Apertura; /* Estado inicial del diálogo */
Repetir
  inFrames  = Lectura(frases del usuario);
  /* inFrames = frames usuario, información semántica */
  input_MD = Adaptar(RVA, inFrames);
  /* input_MD = entradas para transiciones en MD */
  estado_MD = Actualizar(estado_MD, input_MD);
  /* actualización del modelo por turno de usuario */
  RVA = Actualizar(RVA, inFrames);
  /* actualización del registro por turno de usuario */
  estado_MD = Actualizar(estado_MD, RVA);
  /* actualización del modelo por turno del sistema */
  RVA = Actualizar(RVA, output_MD, output_BD);
  /* actualización del registro por turno del sistema */
  outFrames = Adaptar(output_MD);
  /* output_MD = etiqueta asociada al estado alcanzado */
  Escritura(outFrames);
Hasta      estado_MD = Cierre
```

Figura 2: *Algoritmo del gestor de diálogo*

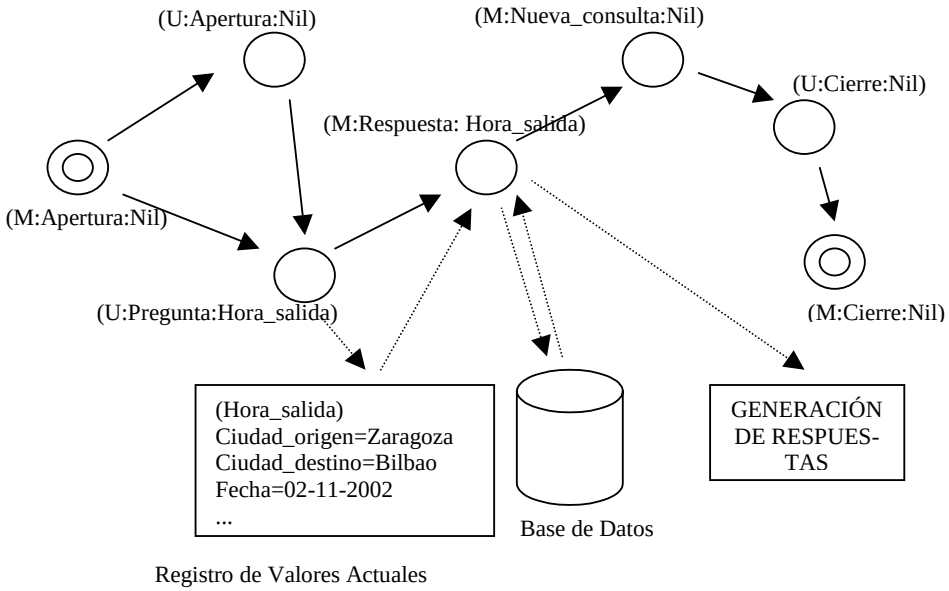

Figura 3: *Ejemplo de una parte del modelo de diálogo*

En la Figura 3 se muestra la interacción del modelo de diálogo con los otros componentes del sistema: la base de datos del sistema de información y el módulo de generación de respuestas.

En las siguientes secciones se exponen los problemas más relevantes encontrados en el desarrollo del sistema y las soluciones que se han adoptado.

5.1 El modelo estocástico de diálogo

A partir de la colección de 215 diálogos etiquetados en el proyecto se han estimado los dos modelos de diálogo. Cada modelo es un autómata de estados finitos estocástico que resulta equivalente a un modelo de bigramas (Bahl, Jelinek y Mercer, 1983) sin suavizar.

El modelo más específico, denominado *modeloNivel2*, contiene 310 estados. El alfabeto para este autómata está formado por un conjunto de cadenas que se ajustan al formato (*Turno:PrimerNivel:SegundoNivel*), donde *Turno* identifica al usuario o al sistema, *PrimerNivel* identifica el tipo de acto de diálogo, y *SegundoNivel* identifica los atributos implicados en ese acto de diálogo.

El modelo más genérico, denominado *modeloNivel1*, contiene 74 estados. Para este autómata el alfabeto está formado por cadenas de la forma (*Turno:PrimerNivel*), representando un menor nivel de detalle de las transiciones posibles en el modelo.

Ambos autómatas serán actualizados en paralelo a lo largo del proceso de diálogo. El autómata *modeloNivel2* constituye el modelo principal, mientras que el autómata *modeloNivel1* puede considerarse un modelo auxiliar. El gestor de diálogo determinará sus respuestas al usuario conforme al modelo principal, pero, cuando no sea capaz de encontrar una transición en el modelo principal, recurrirá al modelo auxiliar para, desde el mismo, forzar la actualización del modelo principal.

Las limitaciones del autómata *modeloNivel2* (derivadas del reducido corpus de aprendizaje) podrían causar que, en un número apreciable de ocasiones, no encontrase una transición. El hecho de que no se actualice el modelo implicaría una falta de respuesta del sistema, y el diálogo en curso quedaría cancelado o bloqueado. La solución que se ha adoptado es realizar un procesamiento previo, *Adaptar (inFrames)*, de la información semántica dada por el usuario, de modo que permita ajustar una entrada al modelo que ofrezca más garantías de éxito. Esta función utiliza las siguientes reglas para generar alternativas a la información proporcionada por el usuario: concatenación, integración y fragmentación.

El siguiente ejemplo ilustra el efecto de tales reglas. Sea *inFrames* la codificación de 3 actos de diálogo (3 *frames* en una intervención del usuario):

inFrames = (Turno:PrimerNivel-A:Atrib1-A,...,AtribN-A)
(Turno:PrimerNivel-B:Atrib1-B,...,AtribN-B)
(Turno:PrimerNivel-C:Atrib1-C,...,AtribN-C)

Algunas de las posibles nuevas cadenas *input_MD* generadas serían:

- a) por concatenación:
(Turno:PrimerNivel-B:Atrib1-B,..., AtribN-B) (Turno:PrimerNivel-C:Atrib1-C,..., AtribN-C)

- b) por integración: cuando en *inFrames* hay dos o más actos de diálogo del mismo tipo (por ejemplo, *PrimerNivel-A = PrimerNivel-B*):
(Turno:PrimerNivel-A:Atrib1-A,...,AtribN-A, Atrib1-B,..., AtribN-B)

- c) por fragmentación: son reducciones de las cadenas correspondientes a los actos de diálogo del usuario, que consisten en generar aleatoriamente una sublista de los atributos especificados en *SegundoNivel*:
(Turno:PrimerNivel-A:Atrib3-A, AtribN-A)
(Turno:PrimerNivel-B:Atrib1-B, Atrib2-B)

Esta activación de múltiples intentos de actualización del autómata se puede considerar equivalente a una especie de suavizado del modelo estadístico. Además, en la función de actualización del modelo, se establece una estrategia de prioridades: primando las cadenas *input_MD* que reflejen con mayor exactitud la información semántica del turno de usuario.

La estrategia expuesta se sintetiza en los siguientes conceptos: creación de un conjunto de alternativas (mediante concatenación, integración y fragmentación) como entradas al modelo, planificación de sus prioridades y relajación progresiva de los criterios de validación de las transiciones en el modelo. La eficacia de esta estrategia se pone de manifiesto en el elevado porcentaje de éxito en la actualización del modelo principal (*modeloNivel2*).

La cobertura del modelo de diálogo se completa con el uso del modelo auxiliar (*modeloNivel1*). Éste es actualizado de modo rutinario, pero su estado actual sólo determina el curso futuro del diálogo cuando fracasa el procedimiento ordinario de actualización del modelo principal.

El procedimiento extraordinario de actualización del modelo principal significa una ruptura radical con su historia precedente, puesto que se prescinde de su estado actual. Esta decisión se hace necesaria debido a que el estado actual del modelo no tiene definida la transición siguiente. En tal caso, el estado actual del modelo auxiliar permite llevar el modelo principal hasta un nuevo estado desde donde proseguir satisfactoriamente el diálogo.

5.2 El registro de valores actuales

El registro de valores actuales (RVA) constituye una memoria donde el sistema almacena información relativa al desarrollo del diálogo, desde su inicio. Así, este registro complementa la información asociada al estado actual, *estado_MD*, que representa sólo los actos de diálogo de la última intervención del usuario. El RVA es un componente esencial para mantener una adecuada actualización del modelo. Esta actualización permitirá generar una respuesta, coherente con el diálogo en curso.

El RVA es actualizado por ambos interlocutores. Los *frames* del usuario contendrán determinados atributos, y en el RVA quedarán registrados esos atributos, así como el momento del diálogo en que se producen. El RVA será consultado por el sistema para decidir su transición en el modelo y para generar su respuesta. El sistema podrá modificar el registro cuando su transición incluya un acto de diálogo de respuesta. Dicha respuesta conlleva la consulta a la base de datos de donde extraerá los nuevos valores de los atributos

El RVA también es decisivo en la generación de las entradas al modelo (*input_MD*) a partir de determinados *frames* de usuario que se caracterizan por la elipsis de la información. En este tipo de *frame* se confirman, afirman o niegan determinados atributos, pero éstos quedan implícitos (han sido citados explícitamente en las precedentes intervenciones y, en una dinámica normal de diálogo, se elude repetirlos). Para tratar estos casos el sistema dispone del RVA para adivinar los atributos implícitos.

Además, el uso del RVA es esencial para discernir entre viajes de ida o viajes de ida y vuelta. La adecuada identificación del tipo de viaje es, quizás, el dato más importante para dirigir adecuadamente los diálogos de la tarea considerada, y, a menudo, este atributo se modifica implícitamente.

5.3 Un ejemplo de diálogo

En la Figura 4 se reproduce un ejemplo de diálogo con el sistema propuesto. En el ejemplo, cada nueva intervención del usuario va precedi-

da de una línea en blanco y un identificador de turno de usuario (U#). Las líneas en cursiva y sangradas indican: las entradas al modelo obtenidas a partir de los *frames* de usuario, las transiciones del modelo por turno de usuario, las transiciones del modelo por turno del sistema, y los *frames* de salida correspondientes al estado alcanzado.

Este ejemplo es ilustrativo de la estrategia de diálogo implementada, pues muestra la actualización del modelo principal por el mecanismo de integración de entradas en el turno *U0*, la actualización desde el modelo auxiliar en los turnos *U1* y *U2*, y la detección implícita del tipo de viaje en el turno *U4*.

U0: Hola buenas, quería viajar de Zaragoza a Bilbao 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?
Entrada(s) Modelo: 0(U:Pregunta:Hora_salida);1(U:Pregunta:Hora_salida_v)
Automata2 actualizado con (U:Pregunta:Hora_salida,Hora_salida_v)
Automata2 actualizado con (S:No_entendido:Nil)(S:Confirmacion:Destino,Dia_salida,Origen)(S:No_entendido:Nil)
S0: outFrame0: (NO-ENTENDIDO); outFrame1: (CONFIRMACION) CONCEPTO:[CIUDAD-DESTINO=Bilbao] [FECHA-SALIDA=02-11-2002] [CIUDAD-ORIGEN=Zaragoza]; outFrame2: (NO-ENTENDIDO)
S0: No le he entendido, ¿quiere viajar de Zaragoza a Bilbao el día dos de noviembre?

U1:
Entrada(s) Modelo: 0(U:Pregunta:Nil)
!!! Automata2 NO actualizado !!! Automata1 actualizado con (U:Pregunta)
Automata2 actualizado desde Automata1 con (U:Pregunta:Hora_salida)
Automata2 actualizado con (S:Confirmacion:Hora_salida)(S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)
S1: outFrame0: (CONFIRMACION) CONCEPTO:[HORA-SALIDA=10.25]; outFrame1: (ESPERA); outFrame2: (INFORMACION) CONCEPTO:[HORA-SALIDA=10.25] [HORA-SALIDA-V=7.05]; outFrame3: (NUEVA-CONSULTA)
S1: Le consulto la hora de salida, un momento, por favor, el tren de ida sale a las 10.25 y el de vuelta a las 7.05, ¿desea algo más?

U2: Sí.
Entrada(s) Modelo: 0(U:Afirmacion:Hora_salida,Hora_salida_v)
!!! Automata2 NO actualizado !!! Automata1 actualizado con (U:Afirmacion)(U:Pregunta)
Automata2 actualizado desde Automata1 con (U:Afirmacion:Tipo_tren)(U:Pregunta:Hora_salida_v)
Automata2 actualizado con (S:Confirmacion:Hora_salida_v)(S:Espera:Nil)(S:Respuesta:Hora_salida_v)(S:Nueva_consulta:Nil)
S2: outFrame0: (CONFIRMACION) CONCEPTO:[HORA-SALIDA-V=7.05]; outFrame1: (ESPERA); outFrame2: (INFORMACION) CONCEPTO:[HORA-SALIDA-V=7.05]; outFrame3: (NUEVA-CONSULTA)
S2: Le consulto la hora de salida, un momento, por favor, el tren de vuelta sale a las 7.05, ¿desea algo más?

U3: ¿Me puede decir a qué hora sale el segundo?
Entrada(s) Modelo: 0(U:Pregunta:Hora_salida)
Automata2 actualizado con (U:Pregunta:Hora_salida)
Automata2 actualizado con (S:Espera:Nil)(S:Respuesta:Hora_salida)(S:Nueva_consulta:Nil)
S3: outFrame0: (ESPERA); outFrame1: (INFORMACION) CONCEPTO:[HORA-SALIDA=12.10]; outFrame2: (NUEVA-CONSULTA)
S3: Un momento, por favor, el tren sale a las 12.10, ¿desea algo más?

U4: Sí, el cinco de noviembre, ¿cuál es el último tren desde Bilbao, para volver?
Entrada(s) Modelo: 0(U:Afirmacion:Hora_salida);1(U:Pregunta:Hora_salida_v)
Automata2 actualizado con (U:Pregunta:Hora_salida_v)
Automata2 actualizado con (S:Confirmacion:Hora_salida_v)(S:Espera:Nil)(S:Respuesta:Hora_salida_v)(S:Nueva_consulta:Nil)
S4: outFrame0: (CONFIRMACION) CONCEPTO:[HORA-SALIDA-V=07.15]; outFrame1: (ESPERA); outFrame2: (INFORMACION) CONCEPTO:[HORA-SALIDA-V=07.15]; outFrame3: (NUEVA-CONSULTA)
S4: Le consulto la hora de salida, un momento, por favor, el tren de vuelta sale a las 7.15, ¿desea algo más?

U5: No muchas gracias, es suficiente.
Entrada(s) Modelo: 0(U:Cierre:Nil)
Automata2 actualizado con (U:Cierre:Nil)
Automata2 actualizado con (S:Cierre:Nil)
S5: outFrame0: (CIERRE)
S5: Gracias por utilizar este servicio, feliz viaje.

Figura 4: *Ejemplo de diálogo*

6 Conclusión

Se ha presentado una implementación de un gestor de diálogo basado en modelos estocásticos. Las pruebas preliminares del sistema correspondientes a intervenciones reales de usuarios, parecen avalar el buen comportamiento de este módulo gestor.

Las limitaciones del modelo estocástico, derivadas de la escasez de muestras de aprendizaje, se han contrarrestado mediante la construcción de un modelo dual, con dos niveles diferentes de abstracción, sin perder su carácter estocástico, así como mediante una estrategia de generalización de la representación semántica de la entrada del usuario y la correspondiente planificación de prioridades y criterios de actualización del modelo. Aún así, el conocimiento sobre la tarea que no aparece reflejado explícitamente en los datos de aprendizaje puede que no sea capturado por los modelos, lo que dificulta el tratamiento de diálogos de iniciativa mixta más complejos. En desarrollos futuros se intentará mejorar la capacidad de generalización de los modelos y se refinará el registro de valores actuales para hacer frente a diálogos más complejos.

Agradecimientos

Este trabajo se ha desarrollado en el marco del proyecto TUSIR subvencionado por la CICYT número TIC2000-0664-C02-01.

Bibliografía

Bahl, L., Jelinek, F. y Mercer, A. 1983. A Maximun Likelihood Approach to Continuous Speech Recognition. *IEEE Trans. on PAMI*, 5(2):179-190.

Bonafonte, A., Aibar, P., Castell, N.n Lleida, E., Mariño, J.B., Sanchis, E. y Torres, I. 2000. Desarrollo de un sistema de diálogo oral en dominios restringidos. En *Primeras Jornadas en Tecnología del Habla*, Sevilla (España).

Cmu communicator spoken dialog toolkit (csdtk). http://www.speech.cs.cmu.edu/communicator/

Córdoba, R., San-Segundo, R., Montero, J.M., Colás, J., Ferreiros, J. Macias-Guarasa, J. y Pardo, J.M. 2001. An interactive directory assistance service for Spanish with large vocabulary recognition. En *Proceedings of the European Conference on Speech Communication and Technology EUROSPEECH*, páginas 1279-1282, Aalborg (Dinamarca).

Glass, J. y Weinstein, E. 2001. Speech builder: facilitating spoken dialog system development. En *Proceedings of the European Conference on Speech Communication and Technology EUROSPEECH*, páginas 1335-1338, Aalborg (Dinamarca).

Lamel, L., Rosset, S., Gauvain, J. L., Bennacef, S., Garnier-Rizet, M. y Prouts, B. 2000. The LIMSI ARISE System. *Speech Communication*, 31(4):339-353.

López-Cozar, R., Rubio, A. J., García, P., Díaz_verdejo, J. E. y López-Soler, J. M. 2000. Sistema telefónico de información a viajeros. En *Primeras Jornadas en Tecnología del Habla*, Sevilla (España). Edición electrónica.

Martínez, C., and Casacuberta F. 2000. A pattern recognition approach to dialog labelling by using finite-state transducers. En *Proceedings of 5th. IberoAmerican Symposium on Pattern Recognition*, pages 669-677, Lisbon (Portugal).

Martínez, C., Sanchis, E., García, F. y Aibar, P. 2002. A labeling proposal to annotate dialogues. En *Proceedings of third International Conference on Language Resources and Evaluation, LREC*, páginas 1577-1582, Las Palmas (España).

Pieraccini, R. y Levin, E. 1997. AMICA: the AT&T Mixed Innitiative Conversational Architecture. En *Proceedings of the European Conference on Speech Communication and Technology EUROSPEECH*, páginas 1875-1878, Rhodes (Grecia).

Sanchis, E. Segarra, E., Galiano, I., García, F. y Hurtado, L. 2000. Modelización de la Comprensión mediante técnicas de Aprendizaje Automático. En *Primeras Jornadas en Tecnología del Habla*, Sevilla (España). Edición electrónica.

Segarra, E. y Sanchis, E., Galiano, M., García, F. y Hurtado, L. 2001. Extracting Semantic Information through Automatic Learning Techniques. En *Proceedings of the IX Spanish Symposium on Pattern Recognition and Image Analysis*, páginas 177-182, Castellón (España).