

# Un método para el tratamiento de situaciones no vistas en la gestión del diálogo\*

David Griol, Lluís F. Hurtado, Emilio Sanchis, Encarna Segarra  
Departament de Sistemes Informàtics i Computació  
Universitat Politècnica de València. E-46022 València, Spain  
{dgriol, lhurtado, esanchis, esegarra}@dsic.upv.es

**Resumen:** El objetivo de este artículo es presentar un método para realizar el tratamiento de las situaciones no vistas en un gestor de diálogo estocástico. El método propuesto aborda el aumento de la cobertura del gestor como un problema de clasificación. Esta aproximación se aplica a un gestor del diálogo estocástico desarrollado para el proyecto DIHANA, cuyo objetivo es el diseño y desarrollo de un sistema de diálogo para el acceso a información sobre trayectos en tren mediante habla espontánea. En el artículo se presentan y evalúan diferentes definiciones de la función que lleva a cabo el proceso de clasificación para el tratamiento de los eventos no vistos.

**Palabras clave:** Sistemas de diálogo hablado, gestión del diálogo, modelos estocásticos, situaciones no vistas, metodologías de clasificación.

**Abstract:** This paper presents an approach to manage unseen situations in a stochastic dialog manager. We model the problem of coverage as a classification problem. This system has been developed in the DIHANA project, whose goal is the design and development of a dialog system to access a railway information system using spontaneous speech in Spanish. We present and evaluate several approaches for the definition of the classification function defined to deal with unseen situations.

**Keywords:** Spoken dialogue systems, dialogue management, stochastic models, unseen situations, classification methodologies.

## 1. Introducción

Durante las últimas décadas, la evolución de las tecnologías del habla ha posibilitado el desarrollo de soluciones basadas en la voz para un conjunto de tareas. Los sistemas de diálogo son un ejemplo de estas soluciones, en las que un computador interactúa con un ser humano para resolver un determinado problema mediante el uso de diálogos hablados. A pesar de que el estado actual de las tecnologías de voz no permite el desarrollo de sistemas de diálogo generales, los sistemas con un dominio restringido se han convertido en un hecho factible durante la última década. Algunos ejemplos pueden encontrarse en (Rudnický et al., 1999), (Zue et al., 2000), (Lamel et al., 2000), (Young, 2002), (Raymond et al., 2004), (López-Cózar et al., 2005) y (Potamianos, Narayanan, y Riccardi, 2005).

Este tipo de sistemas suelen estar compuestos de una serie de módulos que deben cooperar para completar la interacción con el usuario: Reconocedor Automático del Habla, Módulo de Comprensión Semántica, Gestor del Diálogo, Generador de Respuestas, Módulo de Consulta a las Bases de Datos y Sintetizador de texto a voz. Cada uno de estos módulos tiene sus propias características asociadas y la elección del modelo más apropiado para cada uno de ellos depende de una serie de factores: su objetivo, la posibilidad de definir manualmente el comportamiento del módulo, o la capacidad de obtener automáticamente modelos a partir de muestras de entrenamiento.

En el desarrollo del gestor de diálogo para estos sistemas se han introducido en los últimos años aproximaciones estocásticas basadas en datos; tal es el caso de los trabajos de diferentes laboratorios: (Young, 2002), (Levin, Pieraccini, y Eckert, 2000), (Torres et al., 2005) entre otros. Al igual que en el resto de módulos que componen el sistema

\* Este trabajo se ha desarrollado en el marco del proyecto EDECAN subvencionado por la CICYT número TIN2005-08660-C04-02 y del proyecto ACOMP06/053 subvencionado por la GVA.

de diálogo, la aplicación de aproximaciones basadas en corpus depende de la disposición de corpus de la tarea abordada etiquetados convenientemente.

Recientemente, hemos presentado una aproximación estocástica para el desarrollo de un gestor de diálogo (Hurtado et al., 2005). Nuestro gestor de diálogo se basa principalmente en la modelización de las secuencias de actos de diálogo (de usuario y sistema) y la introducción de una partición en el espacio de todas las posibles secuencias de actos de diálogo. Esta partición, que hace posible la estimación del modelo estocástico a partir de los datos de entrenamiento, se define teniendo en cuenta los datos suministrados por el usuario durante la evolución del diálogo. Además, para su definición se tienen en cuenta las medidas de confianza suministradas por los módulos de reconocimiento y comprensión.

Las aproximaciones basadas en la estimación a partir de datos deben tratar el problema de modelar todas las posibles situaciones que pueden ocurrir durante el diálogo (el problema de la cobertura del modelo) utilizando únicamente un corpus de aprendizaje. Es decir, deben tener en cuenta la posibilidad de que el usuario proporcione una situación no prevista en el modelo. En este artículo presentamos un método para adaptar el modelo (gestor del diálogo) a dichas situaciones. Este método propone modelar el problema de la cobertura como un problema de clasificación.

El tratamiento de la cobertura se basa en la realización de una nueva partición en el espacio de secuencias de actos de diálogo posibles. Cada clase de esta partición agrupa todas las secuencias que proporcionan la misma respuesta del sistema. Durante el transcurso de un nuevo diálogo, cuando se genera una secuencia no observada durante el aprendizaje, ésta se clasifica en una de las clases de este conjunto, y la respuesta del sistema en ese momento es la respuesta asociada a la clase seleccionada. La función de clasificación puede definirse de diferentes formas. En este artículo, presentamos cuatro definiciones de esta función: un clasificador naive Bayes multinomial, clasificadores basados en n-gramas, un clasificador basado en la técnica de inferencia gramatical MGGI y un clasificador basado en redes neuronales.

El método propuesto en este trabajo se ha aplicado a la gestión del diálogo en el

marco del proyecto DIHANA (Benedí, Varona, y Lleida, 2004). Este proyecto tiene como uno de sus principales propósitos el diseño y desarrollo de un sistema de diálogo adecuado para el acceso a la información mediante habla espontánea. La tarea del proyecto DIHANA es la consulta en castellano a un sistema de información sobre trayectos de trenes de largo recorrido.

El artículo se estructura de la siguiente forma. La sección 2 describe las características fundamentales y el etiquetado del corpus adquirido para el proyecto DIHANA. La sección 3 presenta el gestor de diálogo estocástico desarrollado. La sección 4 describe las diferentes metodologías de clasificación utilizadas, mostrando la sección 5 los resultados de la evaluación de cada una de ellas.

## 2. *El corpus DIHANA*

En el proyecto DIHANA se adquirió un corpus de 900 diálogos utilizando la técnica del Mago de Oz, en la que una persona emula el comportamiento del gestor de diálogo siguiendo una estrategia prefijada. Participaron 225 usuarios, el número total de turnos de usuario que se adquirieron es 6280, y el vocabulario contiene 823 palabras. Detalles sobre el proceso de adquisición y etiquetado del corpus se pueden encontrar en (Alcácer et al., 2005).

La representación de los turnos de usuario y sistema se realiza en términos de actos de diálogo. Para el caso de los turnos de usuario, los actos de diálogo se corresponden con la interpretación semántica de la intervención del usuario en base a frames (conceptos y atributos). Para la tarea se definieron ocho conceptos (*Hora*, *Precio*, *Tipo-Tren*, *Tiempo-Recorrido*, *Servicios*, *Afirmación*, *Negación*, y *No-Entendido*) y diez atributos (*Origen*, *Destino*, *Fecha-salida*, *Fecha-Llegada*, *Hora-Salida*, *Hora-Llegada*, *Clase*, *Tipo-tren*, *Número-Orden*, y *Servicios*). A continuación, se muestra un ejemplo de la interpretación semántica de un turno de usuario realizada por el módulo de comprensión:

### **Turno del Usuario:**

*Sí, me gustaría saber horarios desde Valencia*

### **Interpretación semántica:**

(Afirmación)

(Hora)

Origen: Valencia

En el caso de los turnos de sistema se definió un etiquetado a tres niveles. El primer nivel contiene las etiquetas referentes a un diálogo general independientemente de la tarea (*Apertura, Cierre, Indefinido, No-Entendido, Espera, Nueva-Consulta, Afirmación, Negación, Pregunta, Confirmación y Respuesta*). El segundo nivel representa los conceptos existentes en el turno, siendo dependiente de la tarea. El tercer nivel representa los valores de los atributos dados en el turno. Las etiquetas definidas para el segundo y tercer nivel son: *Hora-salida, Hora-Llegada, Precio, Tipo-tren, Origen, Destino, Fecha, Número-orden, Número-Trenes, Servicios, Clase, Tipo-Viaje, Tiempo-Recorrido y Nil*. A continuación, se muestra un ejemplo del etiquetado de un turno de sistema:

**Turno del Sistema:**

*¿Quiere tipos de tren a Barcelona?*

**Etiquetado:**

(Confirmación:Tipo-tren:Destino)

**3. El gestor del diálogo estocástico**

Hemos desarrollado un gestor de diálogo basado en la modelización estocástica de las secuencias de actos de diálogo, del usuario y del sistema (Hurtado et al., 2005). El gestor de diálogo estocástico genera turnos de sistema basándose únicamente en la información suministrada por los turnos de usuario y la información contenida en el modelo. Para estimar el modelo estocástico se ha utilizado un corpus de diálogos etiquetados.

Una descripción formal del modelo estocástico propuesto es la siguiente:

Sea  $A_i$  la salida del sistema de diálogo (turno de sistema) en el instante  $i$ , expresada en términos de actos de diálogo. Sea  $U_i$  la representación semántica del turno de usuario (la salida generada por el módulo de comprensión para la intervención del usuario) en el instante  $i$ , expresada en términos de frames. Un diálogo comienza con un turno de sistema que da la bienvenida al usuario y le ofrece sus servicios; llamamos a este turno  $A_1$ . Consideramos que un diálogo es una secuencia de pares (*turno-sistema, turno-usuario*):

$$(A_1, U_1), \dots, (A_i, U_i), \dots, (A_n, U_n)$$

donde  $A_1$  es el turno de bienvenida del sistema, y  $U_n$  es el último turno de usuario. De ahora en adelante, denotaremos el par  $(A_i, U_i)$  como  $S_i$ , el estado de la secuencia del diálogo en el instante  $i$ .

En este contexto, consideramos que, en el instante  $i$ , el objetivo del gestor del diálogo es encontrar la mejor respuesta de sistema  $A_i$ . Esta selección es un proceso local para cada instante  $i$  y tiene en cuenta la secuencia de estados de diálogo que preceden a dicho instante. Esta selección se realiza maximizando:

$$\hat{A}_i = \operatorname{argmax}_{A_i \in \mathcal{A}} P(A_i | S_1, \dots, S_{i-1})$$

donde el conjunto  $\mathcal{A}$  contiene todas las posibles respuestas del sistema. Como el número de posibles secuencias de estados es muy grande, establecemos una partición de las secuencias de estados (por ejemplo, en la historia del diálogo precediendo el instante  $i$ ).

Sea  $DR_i$  el registro del diálogo en el instante  $i$ . El registro del diálogo (*dialog register (DR)*) se define como una estructura de datos que contiene la información sobre los valores de los conceptos y atributos suministrados por el usuario a través de la historia previa del diálogo. Toda la información almacenada en el  $DR_i$  en un instante dado  $i$  es un resumen de la información suministrada por la secuencia  $S_1, \dots, S_{i-1}$ . Cabe destacar que diferentes secuencias de estados pueden conducir al mismo  $DR$ .

Para una secuencia de estados de un diálogo, existe su correspondiente secuencia de  $DR$ :

$$\begin{array}{ccccccc} S_1, & \dots, & S_i, & \dots, & S_n \\ \uparrow & & \uparrow & & \uparrow & & \uparrow \\ DR_1 & & DR_2 & & DR_i & & DR_n \end{array}$$

donde  $DR_1$  contiene la información inicial por defecto del gestor del diálogo (*Origen y Clase*), y los siguientes registros  $DR_i$  se actualizan teniendo en cuenta la información suministrada durante la evolución del diálogo.

Teniendo en cuenta el concepto de  $DR$ , se establece una partición de secuencias de estados de tal modo que: dos secuencias de estados se consideran equivalentes si conducen al mismo  $DR_i$ . De este modo, obtenemos una gran reducción en el número de historias diferentes en los diálogos a expensas de

la pérdida el orden cronológico en el que se suministró la información. Consideramos que ésta es una pérdida menor dado que el orden en el que el usuario suministró la información no es un factor relevante para determinar la próxima respuesta del sistema  $A_i$ .

Tras aplicar las consideraciones anteriores y establecer la relación de equivalencia en las historias de los diálogos, la selección de la mejor  $A_i$  viene dada por:

$$\hat{A}_i = \operatorname{argmax}_{A_i \in \mathcal{A}} P(A_i | DR_{i-1}, S_{i-1})$$

Cada turno de usuario suministra al sistema información relativa a la tarea, es decir, el usuario solicita información sobre un concepto específico y/o suministra los valores de determinados atributos. No obstante, un turno de usuario puede además aportar otros tipos de información, como por ejemplo información independiente de la tarea. Éste es el caso de los turnos correspondientes a los actos de diálogo *Afirmación*, *Negación* y *No-Entendido*. Este tipo de información implica una toma de decisiones diferente a una mera actualización del registro  $DR_{i-1}$ . Por esta razón, para la selección de la mejor respuesta del sistema  $A_i$ , tenemos en cuenta el  $DR$  generado desde el turno 1 al turno  $i - 2$ , y explícitamente consideramos el último estado  $S_{i-1}$ .

Las probabilidades del modelo propuesto se obtienen de un corpus de diálogos etiquetados mediante una estimación por máxima verosimilitud.

### 3.1. Representación del Registro del Diálogo

Para la tarea DIHANA, el  $DR$  se ha definido como una secuencia de 15 campos, cada uno de ellos asociado a un determinado concepto o atributo. La secuencia de campos de conceptos es *Hora*, *Precio*, *Tipo-Tren*, *Tiempo-Recorrido* y *Servicios*. La secuencia de campos de atributos es *Origen*, *Destino*, *Fecha-Salida*, *Fecha-Llegada*, *Hora-Salida*, *Hora-Llegada*, *Clase*, *Tipo-Tren*, *Número-Orden* y *Servicios*.

Para que el gestor de diálogo determine la siguiente respuesta, asumimos que no son significativos los valores exactos de los atributos. Estos valores son importantes para acceder a la base de datos y construir la respuesta del sistema en lenguaje natural. Sin embargo, la

única información necesaria para determinar la siguiente acción del sistema es la presencia o no de conceptos y atributos. Por tanto, la información que almacena el  $DR$  es una codificación de cada uno de sus campos en términos de tres valores,  $\{0, 1, 2\}$ , de acuerdo con el siguiente criterio:

- **0:** El usuario no ha suministrado el concepto o valor del atributo correspondiente.
- **1:** El concepto o atributo está presente con una medida de confianza superior a un umbral prefijado (un valor entre 0 y 1). Las medidas de confianza se generan durante los procesos de reconocimiento y comprensión (García et al., 2003).
- **2:** El concepto o atributo está presente con una medida de confianza inferior al umbral.

De este modo, cada  $DR$  puede representarse como una cadena de longitud 15 cuyos elementos pueden tomar valores del conjunto  $\{0, 1, 2\}$ .

## 4. Tratamiento de las situaciones no vistas

Para conseguir una cobertura total del modelo, deben considerarse las situaciones no vistas, es decir, el gestor del diálogo debe generar una respuesta incluso para los pares  $(DR, S)$  que no se han visto durante la fase de aprendizaje. En este trabajo modelamos el problema de la cobertura como un problema de clasificación

Tal y como se ha comentado en la introducción, en el espacio particionado de las posibles secuencias de actos de diálogo que se ha estimado durante la fase de entrenamiento, se define una segunda partición en clases. Cada clase agrupa todas las secuencias que proporcionan el mismo conjunto de respuestas del sistema. Una vez la fase de entrenamiento ha finalizado, se obtiene un conjunto de clases  $\mathcal{C}$ , de forma que cada clase tiene asociada una respuesta diferente del sistema. Durante el transcurso de un nuevo diálogo, cuando tiene lugar una situación no vista, es decir, cuando el par  $(DR, S)$  no haya sido observado en la fase de entrenamiento, el gestor utilizará una función de clasificación para asociar a la entrada una clase, y por tanto, una respuesta. En este trabajo, presentamos

cuatro aproximaciones para la definición de la función de clasificación, basadas en el aprendizaje de un modelo para cada una de las clases  $c \in \mathcal{C}$ . Estas aproximaciones difieren en la metodología utilizada en el proceso de aprendizaje.

Sea  $(DR, S)$  un par no visto durante el aprendizaje, y  $x$  la cadena que codifica este par. La clasificación se realiza mediante la siguiente maximización:

$$\begin{aligned} \hat{c} &= \operatorname{argmax}_{c \in \mathcal{C}} P(c|x) \\ &= \operatorname{argmax}_{c \in \mathcal{C}} \frac{P(c)P(x|c)}{P(x)} \quad (1) \\ &= \operatorname{argmax}_{c \in \mathcal{C}} P(c)P(x|c) \end{aligned}$$

#### 4.1. Clasificador Multinomial naive Bayes

Los clasificadores naive Bayes se han utilizado frecuentemente en tareas de recuperación de información, clasificación de textos y aprendizaje automático (McCallum, 1999), (Juan y Ney, 2002). En este trabajo, hemos utilizado el clasificador naive Bayes en su modelo de eventos multinomial, estimando sus parámetros a partir de un conjunto de muestras etiquetadas. La clasificación de nuevas muestras se lleva a cabo aplicando la regla de decisión de Bayes, seleccionando la clase que tenga asociada una mayor probabilidad.

La variable  $x$  que aparece en la Ecuación 1 se compone de los siguientes términos:

- Los dos primeros niveles del etiquetado de la última respuesta dada por el sistema ( $A_{i-1}$ ): Esta información se modela mediante una variable multinomial, que posee tantos bits como posibles combinaciones de estos dos niveles (51).

$$\vec{x}_1 = (x_{1_1}, x_{1_2}, x_{1_3}, \dots, x_{1_{51}}) \in \{0, 1\}^{51}$$

- Registro del diálogo ( $DR$ ): Tal y como se ha comentado previamente, el  $DR$  almacena un total de quince características (5 conceptos y 10 atributos). Cada una de estas características pueden tomar los valores  $\{0, 1, 2\}$ . De este modo, cada campo del  $DR$  puede modelarse utilizando una variable multinomial con tres bits.

$$\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in \{0, 1\}^3 \quad i = 2, \dots, 16$$

- Información independiente de la tarea (actos de diálogo *Afirmación*, *Negación*, y *No-Entendido*): Estos tres actos de diálogo se han codificado de forma idéntica a las características almacenadas en el  $DR$ . De esta forma, cada uno de estos tres actos de diálogo puede tomar los valores  $\{0, 1, 2\}$  y modelarse utilizando una variable multinomial con tres bits.

$$\vec{x}_i = (x_{i_1}, x_{i_2}, x_{i_3}) \in \{0, 1\}^3 \quad i = 17, \dots, 19$$

De este modo, la variable  $x$  puede representarse mediante el vector de características:

$$\vec{x} = (\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots, \vec{x}_{19})$$

y su probabilidad viene dada por la expresión:

$$P(\vec{x}) = P(\vec{x}_1)P(\vec{x}_2)P(\vec{x}_3) \dots P(\vec{x}_{19})$$

Las probabilidades de las variables multinomiales se calculan mediante:

$$P(\vec{x}_1) = \prod_{d=1}^{50} p_{1,d}^{x_{1,d}}$$

$$i = 2, \dots, 19 \quad P(\vec{x}_i) = \prod_{d=1}^3 p_{i,d}^{x_{i,d}}$$

donde los coeficientes siguen la expresión:

$$p_{i,d} = \frac{N(x_{i,d} = 1)}{N}$$

siendo  $N$  es el número de muestras de la clase, y  $N(x_{i,d} = 1)$  es el número de muestras de dicha clase con un valor 1 en la posición  $d$ .

#### 4.2. Clasificadores de n-gramas y clasificador MGGI

En estos clasificadores, se estima un autómata de estados finitos para cada una de las clases  $c \in \mathcal{C}$ , a partir de sus correspondientes muestras de entrenamiento. La variable  $x$  contiene la información mostrada para el clasificador multinomial. Cuando sucede una situación no vista durante el transcurso de un

$g(121001200000100) = 1\#12\#21\#30\#40\#51\#62\#70\#80\#90\#100\#110\#121\#130\#140\#15$

Figura 1: Ejemplo de la función de etiquetado  $g$  del MGGI.

	%exacta	%estrategia	%correcta
clasificador bigramas	30.4 %	37.7 %	48.7 %
clasificador trigramas	31.5 %	39.4 %	49.4 %
clasificador MGGI	50.2 %	67.8 %	78.4 %
clasificador multinomial	56.3 %	75.2 %	88.5 %
clasificador MLP	65.6 %	90.9 %	97.1 %

Figura 2: Resultados de la evaluación del DM

diálogo, la próxima respuesta del sistema se selecciona aplicando la Ecuación 1.

Para este trabajo, hemos utilizado tres tipos de clasificadores de estados finitos: modelos de bigramas, modelos de trigramas y modelos MGGI. La metodología MGGI (Morphic Generator Grammatical Inference) (Segarra y Hurtado, 1997) es una técnica de inferencia gramatical que nos permite obtener una variedad de lenguajes regulares. Se basa en la definición de una función de etiquetado  $g$ , es decir, cada símbolo de cada muestra de entrada se especializa (etiqueta) de acuerdo con la función de etiquetado. Diferentes definiciones de  $g$  generarán diferentes modelos.

Definimos la función de etiquetado  $g$  de tal manera que especializa cada campo del  $DR$  agregando información sobre su posición. La Figura 1 muestra un ejemplo de la utilización de la función  $g$  definida para este trabajo. Esta función se aplica a los símbolos procedentes de la representación del  $DR$ , pero no a aquellos pertenecientes a  $S$ .

### 4.3. Clasificador MLP

Los perceptrones multicapa (MLPs) (Rumelhart, Hinton, y Williams, 1986) son las redes neuronales artificiales más comúnmente utilizadas en tareas de clasificación (Castro et al., 2003). Para la aplicación de un MLP en la búsqueda de la respuesta del gestor del diálogo, la capa de entrada recibe la codificación del par de entrada  $(DR_{i-1}, S_{i-1})$ , y la capa de salida se define de acuerdo al número de posibles respuestas del sistema y representa la clase  $c \in \mathcal{C}$  en la cual se clasifica la entrada. El resultado de esta clasificación nos da la correspondiente respuesta del sistema  $A_i$  asociada a dicha clase.

La representación definida para codificar el par de entrada  $(DR_{i-1}, S_{i-1})$  coincide con la mostrada para el clasificador naive Bayes.

## 5. Evaluación

La evaluación y comparación de las diferentes aproximaciones se llevó a cabo mediante un proceso de validación cruzada. El corpus se dividió aleatoriamente en cinco subconjuntos de 1232 muestras (20% del corpus). La evaluación consistió en cinco experimentaciones. En cada una de ellas se utilizó un subconjunto diferente de los cinco definidos como muestras de test, y el 80% del corpus restante se utilizó como partición de entrenamiento.

El número de clases en  $\mathcal{C}$  (es decir, el número de posibles respuestas de sistema) es 51. La media de pares  $(DR, S)$  diferentes en los conjuntos de entrenamiento es 1126. La media de situaciones no vistas es el 14.6% de las muestras de los conjuntos de test.

En el caso de los clasificadores mediante MLP, se utilizó un software desarrollado en nuestros laboratorios. Se extrajo un subconjunto de validación (20%) de cada uno de los conjuntos de test. La topología utilizada fue dos capas ocultas con 110 neuronas cada una.

Para evaluar el funcionamiento de las diferentes metodologías se definieron tres medidas. La primera es el porcentaje de respuestas que coinciden con aquella generada por el Mago de Oz (*%exacta*). La segunda medida es el porcentaje de respuestas que siguen la estrategia definida para la adquisición del corpus DIHANA (*%estrategia*). La tercera medida es el porcentaje de respuestas que son coherentes con el estado actual del diálogo (*%correcta*). Estas dos últimas medidas han sido obtenidas en un proceso de revisión manual. La Figura 2 muestra los resultados obtenidos en la evaluación de las diferentes funciones de clasificación.

Teniendo en cuenta que la estrategia del Mago de Oz posibilita un conjunto de res-

puestas válidas dado un determinado estado del diálogo, los resultados que son relevantes en el ámbito de la gestión del diálogo son *%estrategia* y *%correcta*.

La Figura 2 muestra que los mejores resultados se obtienen a través de la clasificación mediante el MLP. La codificación desarrollada para representar el estado actual del diálogo y el buen funcionamiento del clasificador basado en MLP hacen posible que la respuesta generada por el gestor coincida con una respuesta válida de acuerdo con la estrategia (*%estrategia*) en un porcentaje del 90.9%.

Además, puede observarse que entre los clasificadores basados en modelos de estados finitos, los clasificadores de bigramas y trigramas tienen un peor comportamiento que el clasificador MGGI. Este comportamiento se debe a que estos modelos no pueden capturar dependencias a largo plazo y los modelos que estiman para las distintas clases no discriminan lo suficiente. La función definida para el clasificador MGGI proporciona modelos más discriminantes que los anteriores pero, sin embargo, no consiguen alcanzar las prestaciones del clasificador multinomial y del MLP.

Los clasificadores basados en modelos de lenguaje capturan la información secuencial contenida en la codificación definida; en cambio el resto de los clasificadores tratan esta codificación sin tener en cuenta esta información secuencial, ya que en la codificación definida para el *DR* el orden de los diferentes elementos es irrelevante. Por los resultados obtenidos, parece que estas últimas aproximaciones resultan más adecuadas para el problema planteado.

## 6. Conclusiones

En este artículo, se ha presentado una aproximación para realizar el tratamiento de las situaciones no vistas en un gestor de diálogo. El problema de la cobertura del modelo se ha abordado en términos de una clasificación de las situaciones no vistas. El comportamiento de las diferentes definiciones de la función de clasificación presentadas en el artículo se ha evaluado a través de una experimentación. Los resultados muestran el funcionamiento satisfactorio del método propuesto para el tratamiento de los eventos no vistos, sobre todo cuando se utilizan el clasificador multinomial y el MLP.

Como trabajo futuro caben destacar tres temáticas fundamentales. En primer lugar, la utilización de técnicas de simulación de usuarios con el objetivo de ampliar y mejorar el modelo de diálogo desarrollado a partir del corpus de diálogos simulados. En segundo lugar, la realización de un estudio de errores que nos permita conocer de manera más precisa los fallos cometidos por el modelo. Por último, va a realizarse una evaluación del comportamiento del sistema utilizando usuarios reales con el objetivo de comparar los resultados obtenidos con los presentados en este trabajo.

## Bibliografía

- Alcácer, N., Benedí J. M., Blat F., Granell R., Martínez C. D., y Torres F. 2005. Acquisition and Labelling of a Spontaneous Speech Dialogue Corpus. En *Proc. of the Tenth International Conference Speech and Computer (SPECOM'2005)*, Patras (Grecia).
- Benedí, J.M., A. Varona, y E. Lleida. 2004. DIHANA: Sistema de diálogo para el acceso a la información en habla espontánea en diferentes entornos. En *Actas de las III Jornadas en Tecnología del Habla*, páginas 141–146, Valencia (España).
- Castro, M. J., Vilar D., Sanchis E., y Aibar P. 2003. Uniclass and Multiclass Connectionist Classification of Dialogue Acts. En *Proc. 8th Iberoamerican Congress on Pattern Recognition (CIARP'03)*, volumen 2527 de *LNCS*. Springer-Verlag, páginas 664–673.
- García, F., Hurtado L., Sanchis E., y Segarra E. 2003. The incorporation of Confidence Measures to Language Understanding. En *International Conference on Text Speech and Dialogue (TSD 2003). Lecture Notes in Artificial Intelligence series 2807*, páginas 165–172, Ceské Budejovice (Czech Republic).
- Hurtado, L., Griol D., Sanchis E., y Segarra E. 2005. A stochastic approach to dialog management. En *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'05)*, páginas 226–231.
- Juan, A. y H. Ney. 2002. Reversing and Smoothing the Multinomial Naive Bayes Text Classifier. En *Proc. of PRIS'02*, páginas 200–212, Alacant (Spain).

- Lamel, L., Rosset S., Gauvain J.L., Benaïef S., Garnier-Rizet S., y Prouts B. 2000. The LIMSI ARISE System. *Speech Communication*, 4(31):339–353.
- Levin, E., R. Pieraccini, y W. Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. En *IEEE Transactions on Speech and Audio Processing*, páginas 8(1):11–23.
- López-Cózar, R., Callejas Z., Gea M., y Montoro G. 2005. Multimodal, multilingual and adaptive dialogue system for ubiquitous interaction in educational space. En *Proc. of Applied Spoken Language Interaction in Distributed Environments (ASIDE)*, Aalborg (Denmark).
- McCallum, A. 1999. Multi-Label Text Classification with a Mixture Model Trained by EM. En *Proc. of NIPS'99*.
- Potamianos, A., S. Narayanan, y G. Riccardi. 2005. Adaptive Categorical Understanding for Spoken Dialogue Systems. En *IEEE Transactions on Speech and Audio Processing*, páginas 13(3):321–329.
- Raymond, C., Béchet F., Mori R. D., Damnati G., y Estève Y. 2004. Automatic learning of interpretation strategies for spoken dialogue systems. *Proc. of ICASSP. Montréal, Québec (Canada). vol. 1*, páginas 425–428.
- Rudnicky, A., Thayer E., Constantinides P., Tchou C., Shern R., Lenzo K., Xu W., y Oh A. 1999. Creating natural dialogs in the Carnegie Mellon Communicator system. *Proc. of Eurospeech*, 1(4):1531–1534.
- Rumelhart, D.E., G.E. Hinton, y R.J. Williams, 1986. *PDP: Computational models of cognition and perception, I*, capítulo Learning internal representations by error propagation, páginas 319–362. MIT Press.
- Segarra, E. y L. Hurtado. 1997. Construction of Language Models using Morfic Generator Grammatical Inference MG-GI Methodology. En *Proc. of Eurospeech*, páginas 2695–2698.
- Torres, F., Hurtado L.F., García F., Sanchis E., y Segarra E. 2005. Error handling in a stochastic dialog system through confidence measures. En *Speech Communication*, páginas (45):211–229.
- Young, S. 2002. The Statistical Approach to the Design of Spoken Dialogue Systems. En *Technical Report CUED/F-INFENG/TR.433, Cambridge UK*, páginas 1–25.
- Zue, V., Seneff S., Glass J., Polifroni J., Pao C., Hazen T., y Hetherington L. 2000. Jupiter: A telephone-based conversational interface for weather information. En *IEEE Trans. on Speech and Audio Proc.*, páginas 8(1):85–96.