Procesamiento del Lenguaje Natural, núm. 43 (2009), pp. 113-120 recibido 1-05-2009; aceptado 5-06-2009

# Enhancement of the Input Interface of Spoken Dialogue Systems By Means of Contextual Models and Grammatical Rules

## *Mejora de la Interfaz de Entrada de Sistemas de Diálogo Usando Modelos Contextuales y Reglas Gramaticales*

**Ramón López-Cózar**
Dpto. Lenguajes y Sistemas Informáticos
ETS Informática, Universidad de Granada
rlopezc@ugr.es

**Zoraida Callejas**
Dpto. Lenguajes y Sistemas Informáticos
ETS Informática, Universidad de Granada
zoraida@ugr.es

**Resumen:** Este artículo propone una nueva técnica para mejorar el funcionamiento de la interfaz de entrada de sistemas de diálogo, que presenta una contribución novedosa: el uso de modelos contextuales y reglas gramaticales para corregir errores de reconocimiento automático de habla (RAH). Los experimentos realizados con un sistema de diálogo muestran que la técnica permite incrementar las tasas de exactitud de palabras, comprensión de frases y logro de tareas en 8,5%, 16,54% y 44,17% absoluto, respectivamente.
**Palabras clave:** Sistemas de diálogo, reconocimiento de habla, comprensión de habla.

**Abstract:** This papers proposes a new technique to enhance the performance of the input interface of spoken dialogue systems, which presents a novel contribution: the use of contextual models and grammatical rules to correct errors of automatic speech recognition (ASR). The experiments we carried out employing a spoken dialogue system shows that the technique allows enhancing word accuracy, sentence understanding and task completion by 8,5%, 16,54% y 44,17% absolute, respectively.
**Keywords:** Dialogue systems, speech recognition, spoken language understanding.

## *1 Introducción*

La mayoría de las técnicas disponibles en la literatura para corregir errores de reconocimiento automático de habla (RAH) emplean información estadística acerca de palabras pronunciadas y palabras reconocidas. Por ejemplo, (Ringger y Allen, 1996) es un trabajo clásico en el campo, en el que se propone usar un post-procesador para corregir errores, el cual está constituido por un *canal* y un modelo de lenguaje. El canal modela errores cometidos por el reconocedor, mientras que el modelo de lenguaje proporciona probabilidades acerca de las secuencias de palabras pronunciadas. Los autores entrenaron ambos modelos usando el corpus TRAINS-95, y observaron experimentalmente que gracias al post-procesador se reducía notablemente el número de errores.

Siguiendo un enfoque distinto, (Zhou y Meng, 2004) proponen una técnica que realiza la corrección mediante dos niveles de análisis. En el primero se usa un clasificador para decidir si el resultado del RAH es correcto. En caso de no serlo, el resultado se pasa al segundo nivel, donde se emplea otro clasificador para decidir qué palabras son incorrectas.

En un estudio posterior (Zhou et. al, 2006), se presenta una técnica que realiza la corrección empleando tres niveles de análisis. El primero decide si el resultado de RAH es correcto, el segundo decide qué palabras son incorrectas, y el tercero decide qué caracteres son incorrectos.

Un problema de las técnicas descritas anteriormente, basadas exclusivamente en información estadística, es que necesitan grandes cantidades de datos de entrenamiento.

Este trabajo ha sido realizado en el proyecto HADA TIN2007-64718, financiado por el Ministerio de Educación y Ciencia.

ISSN: 1135-5948 

Además, su éxito depende en gran medida de la calidad de la salida del reconocedor de habla, y también del tamaño de la base de datos de errores usada. Para hacer frente a estos problemas, varios autores han propuesto combinar información léxica, sintáctica y semántica. Por ejemplo, (Jeong et. al., 2004) proponen combinar información léxica y fuentes de información de alto nivel mediante un modelo de lenguaje de máxima entropía.

Basándonos en diversos estudios existentes en la literatura, la técnica que proponemos hace uso de diversos tipos de información para corregir errores de RAH en un sistema de diálogo, concretamente, información léxica, sintáctica, semántica y relacionada con la gestión del diálogo. La novedad de la propuesta reside en que tiene en cuenta, además, modelos contextuales asociados con los tipos de *prompts* que pueden ser generados por el sistema. El modelo usado para realizar la corrección se determina teniendo en cuenta un valor de similitud entre patrones, que se obtiene a partir de frases pronunciadas y patrones aprendidos durante el entrenamiento. Además, la técnica usa reglas gramaticales para corregir algunos errores que no pueden ser detectados usando los modelos contextuales.

## *2 Implementación de la técnica*

Para aplicar la técnica propuesta en un sistema de diálogo, debemos crear *conceptos*, modelos sintáctico-semánticos y modelos léxicos. Asimismo, debemos emplear los algoritmos descritos a continuación.

### 2.1 Conceptos

Definimos *concepto* como un conjunto de palabras clave (*keywords*) de un determinado tipo, necesarias para obtener el significado de frases pronunciadas en un determinado dominio de aplicación de un sistema de diálogo. Por ejemplo, en nuestros experimentos en el dominio del pedido de comida rápida, hemos considerado, entre otros, los siguientes conceptos:

DESEO = {quiero, dame, ponme,…}
NÚMERO = {un, una, uno, dos,…}
COMIDA = {bocadillo, tarta, ensalada,…}
BEBIDA = {agua, cerveza, refresco,…}.

### 2.2 Reglas gramaticales

El formato de las reglas gramaticales usadas por la técnica es el siguiente:

$$pss \rightarrow restricción$$

donde *pss* es un patrón sintáctico-semántico (que se describirá en la siguiente sección) y *restricción* es una condición que debe ser satisfecha por todos los conceptos del patrón. Por ejemplo, una regla que hemos usado en los experimentos es la siguiente:

*pss* →
*número*(NÚMERO) = *número*(BEBIDA) **and**
*número*(BEBIDA) = *número*(TAMAÑO) **and**
*número*(NÚMERO) = *número*(TAMAÑO)

donde *número* es una función que devuelve 'singular' o 'plural' para cada palabra en el concepto recibido como argumento. El objetivo de esta regla es comprobar la correspondencia de número entre las palabras de la frase. Por ejemplo, las palabras de la frase "dos cervezas grandes" satisfacen la restricción de la regla.

### 2.3 Modelos sintáctico-semánticos

Un modelo sintáctico-semántico es una representación conceptual de las frases pronunciadas por los usuarios del sistema en un estado T del diálogo. Este estado está asociado a un tipo de *prompt* del sistema, que representa *prompts* equivalentes cuya finalidad es obtener un determinado tipo de dato. Por ejemplo, en nuestros experimentos, los *prompts*: "¿Algo para comer?", "¿Quieres comer algo?" y "¿Te gustaría algo de comer?" son *prompts* equivalentes asociados al tipo de prompt *preguntar_pedido_comida*.

Para crear un modelo sintáctico-semántico para un estado T del diálogo, debemos transformar cada frase pronunciada por un usuario en lo que denominamos un patrón sintáctico-semántico (*pss*). Dicho patrón es una secuencia de conceptos que se obtiene reemplazando cada palabra de la frase por el/los concepto/s a que pertenece la palabra. Por ejemplo, en nuestros experimentos, el patrón sintáctico-semántico obtenido a partir de la frase: "quiero un bocadillo de jamón y una ensalada verde" es el siguiente:

*pss* = DESEO NÚMERO COMIDA INGREDIENTE NÚMERO COMIDA INGREDIENTE

A partir del análisis de todas las frases pronunciadas en respuesta a cada tipo de prompt, creamos un conjunto de patrones sintáctico-semánticos, en el que eliminamos los patrones repetidos y asociamos a cada patrón único su frecuencia de aparición en el conjunto. Al resultado de este proceso lo denominamos modelo sintáctico-semántico asociado al tipo de prompt T ($MSS_T$).

Llamamos *modelo α* al conjunto de modelos $MSS_T$ creados considerando los *m* tipos de *prompts* generados por un sistema de diálogo:

$$\alpha = \{MSS_{Ti}\}, i = 1 \ldots m.$$

## 2.4 Modelos léxicos

Un modelo léxico ($ML_T$) es una representación del funcionamiento del reconocedor de habla del sistema en un estado T del diálogo.

Para implementar la técnica debemos crear un modelo léxico para cada estado del diálogo. Para ello tenemos en cuenta las frases pronunciadas en respuesta al tipo de prompt, así como los correspondientes resultados de reconocimiento. El formato de este modelo es el siguiente: $ML_T = \{w_a, w_b, p_{ab}\}$, donde $w_a$ es una palabra pronunciada, $w_b$ es la palabra reconocida, y $p_{ab}$ es la probabilidad de obtener $w_b$ dada $w_a$.

Para crear $ML_T$ debemos alinear cada frase pronunciada con la correspondiente frase reconocida, y calcular las probabilidades $p_{ab}$ para cada par de palabras ($w_a$, $w_b$). Para realizar el alineado, en los experimentos hemos utilizado el algoritmo descrito en (Fisher y Fiscus, 1993).

Llamamos modelo β al conjunto de modelos $ML_T$ creados considerando los *m* tipos de *prompts* de un sistema de diálogo:

$$\beta = \{ML_{Ti}\}, \text{con } i = 1 \ldots m.$$

## 2.5 Algoritmos

La corrección de errores de RAH se realiza empleando dos niveles de análisis: estadístico y lingüístico, tal y como se describe a continuación.

### 2.5.1 Corrección a nivel estadístico

El objetivo de la corrección a nivel estadístico es encontrar palabras $w_I$'s en cada frase reconocida que pertenezcan a conceptos incorrectos $K_I$'s. Para cada una de estas palabras, debemos decidir el concepto correcto $K_C$ y seleccionar la palabra más apropiada $w_C \in K_C$ para sustituir a la palabra $w_I$ en la frase reconocida. Este procedimiento se puede implementar en dos pasos:

**Paso 1. Comparación de patrones**. Este paso emplea lo que llamamos un patrón sintáctico-semántico enriquecido ($psse_{INPUT}$), que se obtiene a partir de la frase reconocida. Este patrón consta de una secuencia de *contenedores*, cada uno de los cuales almacena una palabra de la frase. Un contenedor tiene un nombre si la palabra que almacena es una palabra clave. Dicho nombre es el nombre del concepto al cual pertenece la palabra (p.e. DESEO).

El objetivo de este paso es transformar $psse_{INPUT}$ en otro patrón al que llamamos $psse_{BEST}$, que está inicialmente vacío. Para crear este nuevo patrón, creamos un patrón sintáctico-semántico al que llamamos $pss_{INPUT}$, el cual únicamente contiene los conceptos de $psse_{INPUT}$, por ejemplo:

$pss_{INPUT}$ = DESEO NÚMERO COMIDA INGREDIENTE

Seguidamente, comprobamos si $pss_{INPUT}$ coincide con algún patrón existente en el modelo sintáctico-semántico asociado al tipo de prompt T ($MSS_T$). En caso afirmativo, asignamos $psse_{BEST} = psse_{INPUT}$ y continuamos con la corrección a nivel lingüístico (que se discutirá en la sección 2.5.2). En caso negativo, buscamos patrones similares a $pss_{INPUT}$ en $MSS_T$. Para ello, comparamos $pss_{INPUT}$ con cada patrón *p* en el modelo, y calculamos un valor de similitud entre ambos patrones de la siguiente forma:

$$similitud(pss_{INPUT}, p) = (n - m_{ed}) / n$$

donde *n* es el número de conceptos en $pss_{INPUT}$ y $m_{ed}$ es la mínima distancia de edición entre ambos patrones, calculada según el método descrito en (Crestani, 2000).

Llamamos $pss_{SIMILAR}$ a cualquier patrón *p* de $MSS_T$ tal que $similitud(pss_{INPUT}, p) > t$, donde $t \in [0.0, 1.0]$ es un umbral de similitud cuyo valor óptimo debe ser determinado experimentalmente. Consideramos tres casos dependiendo del número de patrones $pss_{SIMILAR}$ existentes:

**Caso 1.** Sólo hay un patrón $pss_{\text{SIMILAR}}$ en $\text{MSS}_\text{T}$. En este caso, creamos un nuevo patrón al que llamamos $pss_{\text{BEST}}$, asignamos $pss_{\text{BEST}} = pss_{\text{SIMILAR}}$ y continuamos con el Paso 2 (Alineamiento de patrones).

**Caso 2**. No hay ningún patrón $pss_{\text{SIMILAR}}$ en $\text{MSS}_\text{T}$. En este caso, tratamos de encontrar algún $pss_{\text{SIMILAR}}$ en el modelo α (descrito al final de la sección 2.3). Si no encontramos ninguno, no realizamos ninguna corrección a nivel estadístico; si sólo encontramos uno, procedemos como se ha explicado en el Caso 1; si encontramos varios, procedemos como se explica en el Caso 3.

**Caso 3.** Hay varios patrones $pss_{\text{SIMILAR}}$ en $\text{MSS}_\text{T}$ o en el modelo α. Por tanto, debemos determinar cual es el mejor. Para ello, buscamos el patrón que tiene mayor valor de similitud con $pss_{\text{INPUT}}$. Si sólo encontramos un $pss_{\text{SIMILAR}}$, asignamos $pss_{\text{BEST}} = pss_{\text{SIMILAR}}$ y continuamos con el Paso 2. Si encontramos varios, seleccionamos aquéllos que tengan mayor frecuencia de aparición en $\text{MSS}_\text{T}$ o en el modelo α (según el modelo que se esté usando) y procedemos de forma análoga. Es decir, si sólo encontramos un $pss_{\text{SIMILAR}}$, asignamos $pss_{\text{BEST}} = pss_{\text{SIMILAR}}$ y continuamos con el Paso 2; si encontramos varios, dada la incertidumbre, no realizamos ninguna corrección a nivel estadístico.

**Paso 2. Alineamiento de patrones**. El objetivo de este paso es construir $psse_{\text{BEST}}$, en caso de que aún esté vacío tras haber realizado el Paso 1. Para ello tenemos en cuenta cada contenedor $C_\text{a}$ de $pss_{\text{INPUT}}$ y consideramos tres casos:

**Caso A**. La palabra $w_\text{a}$ en $C_\text{a}$ no afecta al significado de la frase, es decir, no es una palabra clave, por ejemplo, 'pues'. En este caso, creamos un nuevo contenedor $D$, asignamos $D = C_\text{a}$ y añadimos $D$ a $psse_{\text{BEST}}$.

**Caso B**. La palabra $w_\text{a}$ en $C_\text{a}$ afecta al significado de la frase, es decir, se trata de una palabra clave, por ejemplo, 'bocadillo'. En este caso, debemos determinar si la palabra debe ser corregida, para lo cual tratamos de alinear $C_\text{a}$ con un contenedor $C_\text{b}$ de $pss_{\text{BEST}}$. Consideramos tres casos en función del éxito de la alineación:

**Caso B.1**. $C_\text{a}$ puede ser alineado. En este caso, asumimos que $C_\text{a}$ es correcto y no realizamos ninguna corrección a nivel estadístico. Por tanto, creamos un nuevo contenedor $D$, asignamos $D = C_\text{a}$ y añadimos $D$ a $psse_{\text{BEST}}$.

**Caso B.2**. No es posible alinear $C_\text{a}$. Este caso puede ocurrir en dos situaciones:

**Caso B.2.1**. El contendor $C_\text{a}$ es consecuencia de un error de RAH (inserción), en cuyo caso, descartamos $C_\text{a}$, es decir, éste no se añade a $psse_{\text{BEST}}$.

**Caso B.2.2**. El contenedor $C_\text{a}$ es consecuencia de un error de RAH (substitución). En este caso, debemos encontrar una palabra $w_\text{C}$ para realizar la corrección que pertenezca a un concepto diferente, $w_\text{C} \in C_\text{b}$, almacenar la palabra en un nuevo contenedor $D$, y añadir el contenedor a $psse_{\text{BEST}}$. Para determinar $w_\text{C}$ tenemos en cuenta el modelo léxico asociado al tipo de prompt ($\text{ML}_\text{T}$), y creamos el conjunto $U$ formado por palabras $u \in C_\text{b}$ con las que la palabra $w_\text{I}$ se confunde. Si sólo hay una palabra $u$ en $U$, creamos un nuevo contenedor $D$ al que llamamos $C_\text{b}$, almacenamos en él $u$, y lo añadimos a $psse_{\text{BEST}}$. Si hay varias palabras $u$, realizamos el mismo proceso pero considerando sólo la palabra que tiene mayor probabilidad de confusión con la palabra $w_\text{I}$, en caso de ser ésta única. Si esta palabra no es única, o bien, no hay palabras en $U$, no realizamos ninguna corrección a nivel estadístico.

### 2.5.2 Corrección a nivel lingüístico

El objetivo de la corrección a nivel lingüístico es encontrar palabras significativas erróneamente reconocidas, que no sean detectadas en el nivel estadístico. Por ejemplo, en nuestros experimentos, la frase "una cerveza grande" a veces se reconoce como "dos cerveza grande" dado el acento andaluz de la mayor parte de los usuarios de nuestro sistema de diálogo. Empleando la técnica propuesta, este error no se puede detectar en el nivel estadístico, pues la secuencia de conceptos: NÚMERO BEBIDA TAMAÑO, obtenida a partir de la frase reconocida, es correcta.

Para hacer frente a este problema, usamos el conjunto de reglas gramaticales descritas en la sección 2.2. Para cada regla realizamos el siguiente proceso. El patrón sintáctico-semántico de la regla se coloca en una *ventana*

*deslizante* que se desplaza de izquierda a derecha sobre el patrón $psse_{BEST}$. Si la secuencia de conceptos en la ventana se encuentra en $psse_{BEST}$, aplicamos la *restricción* de la regla a las palabras almacenadas en los contenedores de $psse_{BEST}$. Si las palabras satisfacen la restricción, no realizamos ninguna corrección. En caso contrario, tratamos de determinar la causa que provoca el incumplimiento de la *restricción*, buscando alguna palabra incorrecta. Por ejemplo, aplicando la regla mostrada en la sección 2.2 a la frase "dos cervezas grande" obtendríamos lo siguiente:

*número*('dos') ≠ *número*('cerveza')
*número*('cerveza') = *número*('grande')

lo cual indicaría que 'dos' es la palabra incorrecta.

Para determinar la palabra $w_C$ con la que corregir este error, consideramos el modelo léxico $ML_T$ y creamos el conjunto $U = \{u_1, u_2, ..., u_p\}$ compuesto por palabras que están en el mismo concepto que la palabra incorrecta ('dos', en el ejemplo). Seguidamente, procedemos como se ha descrito en el caso B.2.2, pero teniendo en cuenta que ahora el objetivo no es cambiar una palabra de un concepto por otra palabra de un concepto distinto, sino cambiar una palabra de un concepto (NÚMERO) por otra palabra del mismo concepto.

## *3 Experimentos*

El objetivo de los experimentos ha sido comprobar el funcionamiento de la técnica propuesta usando el sistema de diálogo Saplen (López-Cózar et al., 2006; López-Cózar y Callejas, 2006).

La evaluación se ha realizado considerando las siguientes medidas: exactitud de palabras (*Word Accuracy*, WA), comprensión de frases (*Sentence Understanding*, SU) y logro de tareas (*Task Completion*, TC).

Los experimentos se han realizado empleando dos sistemas de RAH diferentes:

i) *Sistema de RAH de referencia*, compuesto por el reconocedor de habla estándar, basado en HTK, usado por el sistema Saplen.

ii) *Sistema de RAH mejorado*, compuesto por el mismo reconocedor descrito en i) pero teniendo en cuenta, además, un módulo que implementa la técnica propuesta. Este módulo se inserta entre el reconocedor y el módulo de comprensión de habla.

Hemos empleado un corpus de diálogos creado en nuestra Universidad, a partir de la interacción de usuarios (estudiantes) y el sistema Saplen. Este corpus consta de unas 5.500 frases y tiene en torno a 2.000 palabras distintas. A efectos de realizar la evaluación, el corpus ha sido dividido en dos partes disjuntas, una para entrenamiento y otra para test, conteniendo cada una en torno al 50% de las frases.

Usando el corpus de entrenamiento, hemos compilado una bigramática de palabras que permite reconocer frases de los 18 tipos de frases existentes en el corpus, que son los siguientes: pedidos de productos; números de teléfono; códigos postales; direcciones postales; consultas; confirmaciones; cantidades de productos; nombres de comidas; nombres de ingredientes; nombres de bebidas; tamaños; sabores; temperaturas; nombres de calles, avenidas y plazas; números de edificios; identificadores de plantas de edificios; letras de pisos; e indicaciones de error.

Para realizar los experimentos hemos utilizado una técnica de simulación de usuarios desarrollada en un estudio previo (López-Cózar et al., 2006), según la cual, la interacción entre el sistema Saplen y el simulador de usuarios se lleva a cabo utilizando *escenarios*, que representan objetivos de usuarios. Hemos creado dos conjuntos de escenarios, llamados *EscenariosA* (300 escenarios) y *EscenariosB* (100 escenarios).

Cada diálogo generado mediante la técnica de simulación contiene la trascripción ortográfica de cada frase pronunciada, así como el correspondiente resultado de RAH. Dicho diálogo se almacena en un fichero de traza, que se utiliza posteriormente con fines de análisis y evaluación.

Dado que la construcción de los modelos sintáctico-semánticos y léxicos descritos en las secciones 2.3 y 2.4 se ha llevado a cabo empleando diálogos simulados, hemos realizado experimentos adicionales para determinar el número de diálogos necesarios para obtener la máxima cantidad de información sintáctico-semántica y léxica. Los resultados muestran que a partir de 900 diálogos no aumenta la cantidad de información

obtenida. Por tanto, este es el número de diálogos óptimo.

## 3.1 Experimentos usando el sistema de RAH base

Empleando el simulador de usuarios, el sistema Saplen y *EscenariosA*, hemos generado un corpus de 900 diálogos, al que hemos llamado *DiálogosA*$_1$. La Tabla 1 muestra los resultados medios obtenidos al analizar este corpus. Los bajos resultados indican que el sistema Saplen tuvo dificultades para reconocer correctamente palabras (WA) y comprender frases (SU), siendo muy baja, en consecuencia, la tasa de logro de tareas (TC).

El análisis de los ficheros de traza muestra que en algunos casos, las frases reconocidas fueron similares a las pronunciadas por los usuarios. Por ejemplo, la frase "dos fantas grandes de limón" a veces se reconocía como "uno fantas grandes de limón", dado el acento andaluz de la mayoría de los usuarios del sistema.

| WA | SU | TC |
|---|---|---|
| 76,12 | 54,71 | 24,51 |

Tabla 1: Resultados usando el sistema de RAH de referencia (en %).

En otros casos, las frases reconocidas tenían un gran número de errores. Por ejemplo, la frase "quiero una fanta de naranja grande" a veces se reconocía como "queso de manzana tercera".

También se observa en los ficheros de traza que el sistema tuvo problemas para reconocer y comprender correctamente las confirmaciones de los usuarios. En la mayoría de los casos, ello estaba motivado por el acento de los mismos, lo que provocaba que la palabra 'sí' fuera reconocida como 'seis', y la palabra 'no' fuera reconocida como 'dos'.

## 3.2 Experimentos usando el sistema de RAH mejorado

De acuerdo con lo expuesto en la sección 2.1, hemos creado un conjunto de 21 conceptos, que coinciden con los también usados en un estudio previo (López-Cózar y Callejas, 2006).

Siguiendo la propuesta descrita en la sección 2.2, hemos creado un conjunto de reglas gramaticales para comprobar la correspondencia en cuanto a número de los pedidos de comida rápida existentes en el corpus de entrenamiento.

Por otra parte, crear los modelos sintáctico-semánticos y léxicos, discutidos en las secciones 2.3 y 2.4, hemos analizado *DiálogosA*$_1$, obteniendo $\alpha = \{MSS_{Ti}\}$ y $\beta = \{ML_{Ti}\}$, con i = 1 ... 43 pues el sistema Saplen genera 43 tipos de *prompts*.

Para decidir el valor óptimo del umbral de similitud *t* (descrito en la sección 2.5.1), hemos realizado experimentos considerando valores en el rango [0.1 ,0.9]. Empleando el simulador de usuarios y *EscenariosB*, hemos creado un corpus por cada valor de *t* que consta de 300 diálogos, usando en todos los casos la técnica propuesta. El análisis de estos corpora muestra que los mejores resultados de WA, SU y TC se obtienen cuando $t = 0,5$.

Usando este valor óptimo, hemos empleado de nuevo *EscenariosA* para generar otro corpus de 900 diálogos, al que hemos llamado *DialogosA*$_2$. La Tabla 2 muestra los resultados medios obtenidos al analizar este corpus.

| WA | SU | TC |
|---|---|---|
| 84,62 | 71,25 | 68,32 |

Tabla 2: Resultados usando el sistema de RAH mejorado (en %).

Analizando los ficheros de traza observamos que la técnica ha sido útil para corregir diversos tipos de errores de RAH. Por ejemplo, la frase "uno fantas grandes de limón", ha sido corregida sin realizar cambios a nivel sintáctico-semántico, y substituyendo la palabra 'uno' por la palabra 'dos' a nivel léxico, obteniéndose así la frase pronunciada: "dos fantas grandes de limón".

En otros casos, la corrección se ha llevado a cabo realizando cambios a nivel sintáctico-semántico únicamente. Por ejemplo, la frase "una error ensalada de curry" ha sido corregida borrando el concepto ERROR, de acuerdo con lo expuesto en el Caso B.2.1, obteniéndose así la frase pronunciada: "una ensalada de curry".

La técnica también ha permitido hacer frente a los problemas de las confirmaciones, discutidos al final de la sección anterior. Para ello, la técnica reemplaza en primer lugar el concepto NÚMERO por el concepto CONFIRMACIÓN, y en segundo lugar, selecciona la palabra más probable en dicho concepto dada la palabra del concepto NÚMERO. De esta

forma, las frases 'seis' y 'dos' eran corregidas, obteniéndose, respectivamente, 'sí', y 'no'.

También ha sido posible, gracias a la técnica, corregir errores en algunos números de teléfono. Por ejemplo, la frase "dame cinco ocho veintiuno catorce dieciocho", ha sido corregida sustituyendo en primer lugar el concepto ERROR por el concepto NÚMERO, y en segundo lugar, seleccionando la palabra más probable de este último concepto dada la palabra 'dame'. De esta forma, se ha obtenido la frase pronunciada: "nueve cinco ocho veintiuno catorce dieciocho".

Asimismo, ha sido posible corregir errores en el reconocimiento de algunos códigos postales. Por ejemplo, la frase "dieciocho cero cero pavo" ha sido corregida sustituyendo el concepto INGREDIENTE por el concepto NÚMERO, y seguidamente, seleccionando la palabra más probable en éste último concepto dada la palabra 'pavo', obteniéndose así la frase pronunciada: "dieciocho cero cero uno".

La técnica también ha sido útil para corregir errores en el reconocimiento de algunas direcciones postales. Por ejemplo, la frase "calle almona del boqueron error cinco segundo cero" ha sido corregida realizando dos acciones. En primer lugar, reemplazando el concepto ERROR por el concepto ID_NUMERO, y buscando la palabra más probable en éste último concepto dada la palabra 'error'. En segundo lugar, reemplazando el concepto NÚMERO por el concepto LETRA y buscando la palabra más probable en éste último dada la palabra 'cero'. De esta forma, se ha obtenido la frase pronunciada: "calle almona del boquerón número cinco segundo letra h".

Ha habido diversos casos en que la técnica no ha llegado a detectar errores, y por tanto no los ha corregido. Esto ha ocurrido cuando debido a los errores, algunas palabras en las frases reconocidas han sido substituidas por otras palabras, y el resultado ha sido válido en nuestro dominio de aplicación (pedido de comida rápida). Por ejemplo, la frase "dos ensaladas verdes" algunas veces se reconocía como "doce ensaladas verdes". Para esta frase reconocida la técnica no ha detectado ningún error, pues el patrón sintáctico-semántico obtenido a partir de ella coincidía con un patrón aprendido, y además, las palabras concordaban en cuanto a número.

### 3.2.1 Ventajas derivadas del empleo de los modelos $MSS_T$ y α, así como del umbral *t*

El objetivo de este experimento ha sido comprobar si la estrategia propuesta en la sección 2.5.1, es decir, usar los modelos $MSS_T$ o α en función del valor del umbral de similitud *t*, es preferible a usar dos estrategias alternativas:

i) Usar sólo el modelo α, sin tener en cuenta los modelos $MSS_T$.

ii) Usar los modelos $MSS_T$, y si el patrón $pss_{INPUT}$ no se encuentra en ninguno de estos modelos, usar el modelo α, pero sin tener en cuenta el valor de *t*.

El modelo α usado ha sido el mismo que se ha creado empleando $Diálogos A_1$, siendo $t = 0{,}5$.

En primer lugar, hemos implementado la estrategia i), y usando *EscenariosA*, hemos generado otro corpus de 900 diálogos al que hemos llamado $Diálogos A_3$.

Seguidamente, hemos implementado la estrategia ii), y usando de nuevo *EscenariosA*, hemos generado otro corpus de 900 diálogos al que hemos denominado $Diálogos A_4$.

Por consiguiente, $Diálogos A_1$, $Diálogos A_3$ y $Diálogos A_4$ han sido creados usando el mismo conjunto de escenarios, y todos ellos tienen el mismo número de diálogos. La única diferencia reside en la estrategia utilizada para determinar el modelo sintáctico-semántico ($MSS_T$ o α) a usar para realizar la corrección. La Tabla 3 muestra los resultados medios obtenidos al analizar los copora $Diálogos A_3$ y $Diálogos A_4$.

| Corpus | WA | SU | TC |
|---|---|---|---|
| $Diálogos A_3$ | 80,5 | 61,67 | 39,78 |
| $Diálogos A_4$ | 82,26 | 66,84 | 55,35 |

Tabla 3: Resultados empleando estrategias alternativas para seleccionar el modelo sintáctico-semántico (en %).

En análisis de los ficheros de traza muestra que la estrategia utilizada para seleccionar el modelo a usar ($MSS_T$ o α), influye en gran medida en el éxito de la corrección de los errores de las confirmaciones. Así, cuando siempre se usa $MSS_T$, la corrección se realiza de forma correcta en la mayor parte de los casos; sin embargo, si siempre usa α, la corrección suele realizarse de forma incorrecta.

### 3.2.2 Ventajas derivadas del empleo de los modelos $ML_T$ y β, así como del umbral *t*

En este experimento hemos comprobado si usar los modelos $ML_T$ o β en función del valor de *t*, según se propone en la sección 2.5.1, es preferible a usar β siempre. Para ello, hemos usado el modelo β creado con el corpus *DiálogosA*$_1$. Empleando de nuevo *EscenariosA*, hemos generado otro corpus que contiene 900 diálogos, al que hemos llamado *DiálogosA*$_5$. Por consiguiente, *DiálogosA*$_1$ y *DiálogosA*$_5$ han sido obtenidos usando el mismo conjunto de escenarios, y ambos cuentan con el mismo número de diálogos, siendo la única diferencia entre ambos la estrategia utilizada para seleccionar β. La Tabla 4 muestra los resultados medios obtenidos al analizar *DiálogosA*$_5$.

| Corpus | WA | SU | TC |
|---|---|---|---|
| *DiálogosA*$_5$ | 81,40 | 65,61 | 60,89 |

Tabla 4: Resultados empleando otra estrategia para seleccionar el modelo léxico (en %).

El análisis de los ficheros de traza muestra que las probabilidades de confusión de las palabras no son las mismas en los modelos $ML_T$ y β. Por ejemplo, según este último modelo, la mayor probabilidad de confundir la palabra 'error' con una palabra del concepto NÚMERO es 0,0370, y dicha palabra es 'dieciséis'. Sin embargo, considerando el modelo $ML_{T=PEDIDO\_PRODUCTO}$, esta probabilidad es 0,0090, y la palabra es 'una'. Por tanto, la palabra usada para realizar la corrección es 'dieciséis' si se usa β, y es 'una' si se usa $ML_{T=PEDIDO\_PRODUCTO}$, lo cual es determinante para realizar la corrección adecuada.

## *4 Conclusiones y trabajo futuro*

Comparando los resultados mostrados en las Tablas 1 y 2, se observa que la técnica propuesta ha permitido mejorar el funcionamiento del sistema Saplen en términos de exactitud de palabras (WA), comprensión de frases (SU) y logro de tareas (TC) en un 8,5%, 16,54% y 44.17% absolutos, respectivamente.

Además, la comparación de las Tablas 2, 3 y 4 muestra que la estrategia propuesta para seleccionar los modelos que debemos usar durante el proceso de corrección, basada en tipos de *prompts* y un umbral de similitud, es preferible a usar estrategias alternativas.

En trabajos futuros tenemos previsto tener en cuenta fuentes de información adicionales para intentar detectar errores de RAH que en la implementación actual no pueden ser detectados. Por ejemplo, podríamos considerar conocimiento experto, específico del dominio de aplicación del sistema de diálogo. De esta forma, la frase "doce ensaladas verdes" sería considerada susceptible de contener algún error, pues es inusual que los usuarios pidan una cantidad tal elevada de un determinado producto.

Asimismo, tenemos previsto estudiar el comportamiento de la técnica usando diversos umbrales de similitud, uno por cada tipo de *prompt*, en lugar de usar un solo umbral independiente del tipo de *prompt*.

## *Bibliografía*

Crestani, F. 2000. Word recognition errors and relevance feedback in spoken query processing. Proc. of Conf. on Flexible Query Answering Systems, pp. 267-281.

Fisher, W. M., Fiscus, J. G. 1993. Better alignment procedures for speech recognition evaluation", Proc. of ICASSP, pp. 59-62.

Jeong, M., Jung, S., Lee, G. G. 2004. Speech recognition error correction using maximum entropy language model. Proc. of Interspeech, pp. 2137-2140.

López-Cózar, R., Callejas, Z., McTear, M. 2006 Testing the performance of a spoken dialogue system by means of a new artificially simulated user, *Artificial Intelligence Review*, 26, pp. 291-323.

López-Cózar, R., Callejas, Z. 2006. Combining language models in the input interface of a spoken dialogue system, *Computer Speech and Language*, 20, pp. 420-440.

Ringer, E. K., Allen, J. F. 2000. A fertility model for post correction of continuous speech recognition. Proc. of ANLP-NAACL Satellite Workshop, pp. 1-6.

Zhou, Z., Meng, H. 2004. A two-level schemata for detecting recognition errors. Proc. of ICSLP 2004, pp. 449-452

Zhou, Z., Meng, H., Lo, W. K. 2006. A multi-pass error detection and correction framework for Mandarin LVCSR, Proc. of ICSLP, pp. 1646-1649.