NUEVA TÉCNICA DE POST-CORRECCIÓN DE ERRORES DE RAH PARA SISTEMAS DE DIÁLOGO ORAL

Ramón López-Cózar, Zoraida Callejas

Dpto. de Lenguajes y Sistemas Informáticos, E.T.S.I. Informática y de Telecomunicación, Universidad de Granada, 18071 Granada, {rlopezc, zoraida}@ugr.es

RESUMEN

Este artículo propone una técnica para corregir errores de RAH en sistemas de diálogo oral que presenta dos novedades. Por una parte, el uso de varios contextos en los que se puede corregir un error de RAH, y por otra, el uso de valores de confianza asignados a las palabras empleadas para corregir palabras erróneas. Los resultados experimentales obtenidos usando el sistema de diálogo Saplen muestran que la técnica permite mejorar las tasas de exactitud de palabras, comprensión de frases, recuperación implícita y logro de tareas en 8,5%, 16,54%, 4% y 43,81% absoluto, respectivamente.

1. INTRODUCCIÓN

La mayor parte de las técnicas de post-corrección de errores de RAH existentes en la literatura usan información estadística acerca de palabras pronunciadas y palabras reconocidas [1], [2]. No obstante, estas técnicas requieren grandes cantidades de datos de entrenamiento. Además, el éxito de las mismas depende de la calidad de los resultados de RAH, y del tamaño de la base de datos de errores usada en el aprendizaje. Para soslayar estos problemas, diversos autores proponen usar información léxica, sintáctica, semántica o relacionada con la historia del diálogo [3].

La técnica que proponemos, aplicable a sistemas de diálogo oral, sigue esta última aproximación. Además de considerar diversas fuentes de información, tiene en cuenta diversos contextos en los cuales se pueden corregir errores de RAH. Asimismo, propone un método simple para asignar un valor de confianza a cada palabra empleada para corregir otra errónea. La técnica toma cada frase reconocida, y realiza correcciones en dicha frase. Se asume que el reconocedor asigna un valor de confianza a cada palabra reconocida, p.e. "quiero (0,7590) un (0,9268) bocadillo (0,8182) de (0,6532) lomo (0,4598)". No obstante, la técnica es igualmente aplicable si el reconocedor no proporciona dichos valores, descartándose en tal caso el algoritmo para el cálculo de los mismos.

Este trabajo ha sido subvencionado por el proyecto HADA TIN2007-64718 (Ministerio de Educación y Ciencia).

2. LA TÉCNICA PROPUESTA

2.1 Elementos necesarios

2.1.1 Clases de palabras

La técnica usa clases de palabras K_i que han de ser creadas a partir de las transcripciones de un corpus de frases de entrenamiento. Cada clase contiene palabras de un determinado tipo que son significativas para obtener el contenido semántico de las frases. Por ejemplo, clases de palabras relacionadas con el "pedido de comida rápida" son las siguientes: DESEO = {quiero, dame, ponme,…}, CANTIDAD = {un, una, uno, dos,…}, COMIDA = {bocadillo, tarda, ensalada,…}, BEBIDA = {agua, cerveza, refresco,…}. Llamamos Ω al conjunto de clases de palabras usadas para implementar la técnica propuesta en un determinado dominio de aplicación: $\Omega = \{K_1, K_2, \ldots, K_r\}$.

2.1.2 Reglas gramaticales

La técnica emplea un conjunto de reglas gramaticales simples que se usan para corregir errores de RAH que afectan a la semántica de las frases reconocidas. El formato general de una regla gramatical r_i es el siguiente: r_i: $pss_i \rightarrow restricción_i$, donde pss_i es un patrón sintáctico-semántico (descrito en la Sección 2.1.3) y $restricción_i$ es una condición que debe ser satisfecha por todas las clases de palabras en pss_i. Por ejemplo, una regla gramatical usada en nuestros experimentos es la siguiente:

r_1 : pss_1 → número(CANTIDAD) = número(BEBIDA)
and número(BEBIDA) = número(TAMAÑO)
and número(CANTIDAD) = número(TAMAÑO)

donde número es una función que devuelve 'singular' o 'plural' para cada palabra en la clase de palabras que recibe como entrada, y pss_1 es el patrón: CANTIDAD BEBIDA TAMAÑO.

2.1.3 Modelos sintáctico-semánticos

Un modelo sintáctico-semántico es una representación de la estructura conceptual de un tipo de frases

pronunciadas por locutores en un determinado dominio de aplicación. Por ejemplo, en nuestros experimentos hemos creado modelos sintáctico-semánticos para pedidos de comida y/o bebida, números de teléfono, códigos postales, direcciones, confirmaciones, etc. Para crear un modelo, se toma la trascripción de cada frase de un determinado tipo y se transforma en lo que llamamos un *patrón sintáctico-semántico* (*pss*), que es una secuencia de las clases de palabras en la trascripción. Por ejemplo, el *pss* correspondiente a la trascripción: *"por favor, quiero un bocadillo de jamón y una ensalada verde"* es el siguiente:

pss = DESEO CANTIDAD COMIDA INGREDIENTE CANTIDAD COMIDA INGREDIENTE

A partir del análisis de todas las transcripciones se crea un conjunto de *pss*'s. Dicho conjunto debe ser procesado para eliminar *pss*'s repetidos, y asociar a cada *pss* su frecuencia de aparición en el conjunto. Al resultado de este proceso lo denominamos *modelo sintáctico-semántico* asociado con el prompt T (MSS_T). Si el sistema de diálogo en que se pretende aplicar la técnica propuesta genera u tipos de prompts distintos, debemos crear u modelos sintáctico-semánticos. Llamamos α al conjunto formado por todos los modelos sintáctico-semánticos creados para el sistema de diálogo: $\alpha = \{MSS_{Ti}\}$, i = 1 ... u.

2.1.4 Modelos léxicos

Los modelos léxicos contienen información acerca del funcionamiento del reconocedor de habla del sistema de diálogo para cada estado del diálogo. Se asume que existe un estado del diálogo por cada tipo de prompt generado por el sistema. La técnica propuesta requiere que se cree un modelo léxico para cada tipo de prompt T (ML_T), cuyo formato es el siguiente: $ML_T = \{w_i, w_j, p_{ij}\}$, donde w_i es una palabra pronunciada por un locutor, w_j es el resultado de reconocimiento correspondiente a w_i, y p_{ij} es la probabilidad a posteriori de obtener w_j dada w_i. Para crear un ML_T se debe alinear cada trascripción de frase pronunciada con su correspondiente frase reconocida, y calcular la probabilidad p_{ij} para cada par de palabras (w_i, w_j). Si el sistema de diálogo en que pretendemos aplicar la técnica genera u tipos de prompts distintos, debemos crear u modelos léxicos. Llamamos β al conjunto formado por todos los modelos léxicos creados para el sistema: $\beta = \{ML_{Ti}\}$, i = 1 ... u. Para crear dichos modelos, hemos usado el algoritmo de alineación descrito en [4].

2.2 Algoritmos para implementar la técnica

2.2.1 Cálculo de valores de confianza y corrección a nivel estadístico

El objetivo de la corrección a nivel estadístico es encontrar palabras *w*'s en la frase reconocida que pertenezcan a conceptos incorrectos *K*'s. Si se encuentra una de estas palabras, la técnica debe determinar el concepto correcto *K*' y seleccionar la palabra más adecuada $w' \in K'$ para sustituir a la palabra w en la frase reconocida. Asimismo, la técnica debe calcular un valor de confianza para *w*', denotado C(*w*'), en caso de que la palabra w tuviera asociado un valor de confianza. Para calcular dicho valor, se tiene en cuenta el número de palabras existentes en el modelo léxico empleado para realizar la corrección, es decir, palabras u_i con las cuales se confunde la palabra w. Supongamos que dichas palabras forman el conjunto $U = \{u_1, u_2, \ldots, u_p\}$. Si U sólo contiene una palabra, u_1, entonces $w' = u_1$ y C(w') = 1,0. Si U contiene varias palabras, entonces w' es la palabra con mayor probabilidad de confusión con w. Si denotamos como p dicha probabilidad, entonces, C(w') = p. El algoritmo para realizar la corrección a nivel estadístico emplea los dos pasos siguientes:

Paso 1. ***Comparación de patrones***. Este paso opera sobre un patrón sintáctico-semántico *enriquecido* obtenido a partir de la frase reconocida, al que llamamos $psse_{INPUT}$. Dicho patrón es una secuencia de *contenedores* que almacenan información acerca de las palabras y valores de confianza en la frase reconocida, así como de las clases a las que pertenecen las palabras. La finalidad de este paso es transformar $psse_{INPUT}$ en otro patrón llamado $psse_{BEST}$. Para ello, se crea inicialmente un patrón sintáctico-semántico llamado pss_{INPUT}, que contiene únicamente las clases de palabras en $psse_{INPUT}$, por ejemplo:

pss_{INPUT} = DESEO CANTIDAD COMIDA INGREDIENTE

Seguidamente, se determina si pss_{INPUT} coincide con alguno de los patrones en MSS_T. En caso afirmativo, se asigna $psse_{BEST} = psse_{INPUT}$ y se prosigue con la corrección a nivel lingüístico (Sección 2.2.2). En caso contrario, se buscan patrones similares a pss_{INPUT} en MSS_T. Para ello, se compara pss_{INPUT} con cada patrón p en MSS_T, calculando un valor de similitud entre ambos patrones como sigue: similitud(pss_{INPUT}, p) = (n – m_{ed}) / n, donde n es el número de clases de palabras en pss_{INPUT}, y m_{ed} es la distancia mínima de edición entre ambos patrones [5]. La técnica selecciona como similares aquellos patrones con un valor de similitud mayor que un umbral $t \in [0{,}0–1{,}0]$, cuyo valor óptimo debe ser calculado empíricamente. Llamamos $pss_{SIMILAR}$ a cualquier patrón p en MSS_T tal que similitud($pss_{SIMILAR}$, p) > t. Consideramos 3 casos:

Caso 1. Sólo hay un $pss_{SIMILAR}$ en MSS_T. En este caso, el algoritmo crea un nuevo patrón llamado pss_{BEST}, realiza la asignación $pss_{BEST} = pss_{SIMILAR}$, y prosigue con el Paso 2 (Alineamiento de patrones).

Caso 2. No hay ningún $pss_{SIMILAR}$ en MSS_T. En este caso, se intenta encontrar algún $pss_{SIMILAR}$ en el conjunto α (descrito en la Sección 2.1.3). Si no se encuentra ninguno, no se realiza corrección a nivel estadístico. Si sólo se

encuentra uno, el procesamiento es como en el Caso 1. Si se encuentra más de uno, el procesamiento es como en el Caso 3.

Caso 3. Existen varios pss_{SIMILAR}'s en MSS_T (o en α). La cuestión entonces es determinar el mejor pss_{SIMILAR}. Para ello se selecciona el pss_{SIMILAR} que tenga mayor similitud con pss_{INPUT}. Si sólo existe un pss_{SIMILAR}, se asigna $pss_{\text{BEST}} = pss_{\text{SIMILAR}}$ y se prosigue en el Paso 2. Si existen varios pss_{SIMILAR}'s, se seleccionan aquéllos que tengan mayor frecuencia de aparición en MSS_T (o en α). Si sólo existe uno, se asigna $pss_{\text{BEST}} = pss_{\text{SIMILAR}}$ y se prosigue en el Paso 2. En caso de existir más de uno, no se realiza corrección a nivel estadístico.

Paso 2. ***Alineamiento de patrones***. Llegados a este punto, se ha obtenido pss_{BEST} a partir de pss_{INPUT}, pero no se ha creado $psse_{\text{BEST}}$. El objetivo de este paso es crear este patrón. Para ello, se alinea pss_{INPUT} con $psse_{\text{BEST}}$, y considerando cada contenedor C_i en pss_{INPUT}, se analizan 3 casos:

Caso A. La palabra w_i en C_i no afecta al contenido semántico de la frase. En este caso, se crea un nuevo contenedor D_i, se asigna $D_i = C_i$ y se añade D_i a $psse_{\text{BEST}}$.

Caso B. La palabra w_i en C_i afecta al contenido semántico de la frase. En este caso, se estudia si dicha palabra debe ser corregida, teniendo en cuenta los *pss*'s observados en el entrenamiento. Para ello, se intenta alinear el concepto C_i con algún concepto C_j en pss_{BEST}, y se consideran 3 casos:

Caso B.1. $C_i \neq C_j$. Este caso representa la situación en que se encuentra un concepto erróneo en la frase reconocida. Por tanto, se debe encontrar una palabra $w' \in C_j$, determinar su valor de confianza, almacenar dicha información en un nuevo contendor D_i, y añadir este contendor a $psse_{\text{BEST}}$. Para encontrar w' se usa ML_T y se crea el conjunto U de palabras $u \in C_j$ con las cuales se confunde la palabra w. Si sólo existe una palabra u_1 en U, se crea un nuevo contendor D_i cuyo nombre es el del contenedor C_j, y que contiene a la palabra u_1 junto con su valor de confianza, $C(u_1) = 1.0$. Finalmente, se añade D_i a $psse_{\text{BEST}}$. Si U está vacío, se actúa de forma análoga, pero considerando el conjunto β en lugar de ML_T. Si existe más de una palabra en U, se actúa de forma análoga, seleccionando la palabras que tenga mayor probabilidad de confusión con la palabra w.

Caso B.2. $C_i = C_j$. En este caso, se asume que el concepto de la frase reconocida es correcto, y que por tanto, no se debe realizar ninguna corrección. Por consiguiente, se hace $D_i = C_i$ y se añade D_i a $psse_{\text{BEST}}$.

Caso B.3. No es posible alinear C_i. Esto ocurre cuando el concepto C_i proviene de una palabra insertada en la frase reconocida a causa de un error de RAH. El algoritmo descarta C_i, es decir, no lo añade a $psse_{\text{BEST}}$.

2.2.2 Corrección a nivel lingüístico

La finalidad de la corrección a nivel lingüístico es corregir errores no detectados a nivel estadístico que afectan al contenido semántico de las frases. Por ejemplo, en nuestros experimentos, la frase "*una cerveza grande*" a veces es reconocida como "*dos cerveza grande*". Este tipo de error no puede ser reconocido a nivel estadístico, pues la secuencia de conceptos en la frase es correcta. Para realizar la corrección, se usa el conjunto de reglas gramaticales descrito en la Sección 2.1.2. Para cada regla se realiza el siguiente procesamiento. El patrón sintáctico-semántico de la misma se introduce en una *ventana* que se *desliza* de izquierda a derecha sobre $psse_{\text{BEST}}$. Si la secuencia de conceptos en la ventana se encuentra en $psse_{\text{BEST}}$, entonces se aplica la restricción $restricción_i$ a las palabras existentes en los contadores de $psse_{\text{BEST}}$. Si las palabras cumplen la restricción, no se realiza ninguna corrección. En caso contrario, se intenta determinar la causa de la incongruencia, buscando una palabra incorrecta. Este es el caso del ejemplo, pues número(CANTIDAD) ≠ número(BEBIDA). Para determinar la palabra w' con que se debe sustituir la palabra considerada errónea w, se examina el modelo léxico LM_T y se define el conjunto $U = \{u_1, u_2, ..., u_p\}$, constituido por palabras pertenecientes a la misma clase de palabras que w. Seguidamente, se actúa de forma análoga a como se ha explicado en el Caso B.1, con la salvedad de que ahora el objetivo no es reemplazar un concepto por otro, sino una palabra de un concepto por otra palabra del mismo concepto.

3. EXPERIMENTOS

El objetivo de los experimentos ha sido comprobar la efectividad de la técnica propuesta usando el sistema de diálogo Saplen [6], [7]. Las medidas de evaluación han sido las siguientes: exactitud de palabras (WA), comprensión de frases (SU), recuperación implícita (IR) y logro de tareas [8]. Para realizar comparaciones, los resultados experimentales han sido obtenidos usando dos sistemas de RAH distintos:

i) Sistema de RAH *base*, formado únicamente por el reconocedor de habla basado en HTK usado inicialmente por el sistema Saplen.
ii) Sistema de RAH *mejorado*, compuesto por el mismo reconocedor basado en HTK, seguido de un módulo adicional que implementa la técnica propuesta.

Se ha usado un corpus de frases construido a partir de diálogos reales entre estudiantes de nuestra Universidad y el sistema Saplen. Dicho corpus consta de unas 5.500 frases y unas 2.000 palabras distintas. El corpus ha sido dividido en dos corpus disjuntos, uno para entrenamiento (2.750 frases) y el otro para evaluación (2.750 frases). Usando el corpus de entrenamiento, se ha compilado una bigramática de palabras que permite reconocer frases de los 18 tipos existentes en el corpus.

Los experimentos han sido realizados empleando un simulador de usuarios desarrollado en un trabajo previo [6]. La interacción entre el sistema Saplen y el simulador se lleva a cabo empleando escenarios que representan objetivos que el simulador debe intentar conseguir durante la interacción. Hemos creados dos corpora de escenarios: *EscenariosA* (300 escenarios) y *EscenariosB* (100 escenarios). Cada diálogo generado durante la interacción simulador-Saplen se ha almacenado en un fichero log que se ha analizado para obtener los resultados experimentales.

Dado que la creación de los modelos sintáctico-semánticos y léxicos descritos en las Secciones 2.1.3 y 2.1.4 se ha realizado empleando los diálogos obtenidos mediante el simulador de usuarios, hemos realizado estudios preliminares para determinar la cantidad de diálogos que permite obtener la mayor cantidad posible de información sintáctico-semántica y léxica. Los resultados obtenidos muestran que a partir de 900 diálogos no aumenta la cantidad de información aprendida.

3.1 Experimentos con el sistema de RAH base

Usando el simulador de usuarios y empleando *EscenariosA* hemos generado otro corpus que contiene 900 diálogos. La Tabla 1 muestra los resultados medios (en %) obtenidos a partir de la evaluación de dicho corpus.

WA	SU	IR	TC
76,12	54,71	9,19	24,51

Tabla 1. Resultados de evaluación usando el sistema de RAH *base*

3.2 Experimentos con el sistema de RAH mejorado

De acuerdo con lo expuesto en la Sección 2.1.1, se ha creado un conjunto de clases de palabras $\Omega = \{K_1, K_2,\ldots, K_{21}\}$, que contiene las mismas clases usadas en un estudio previo [7]. Teniendo en cuenta lo expuesto en la Sección 2.1.2, se ha creado un conjunto de reglas gramaticales para determinar la concordancia en cuanto a número en las frases de tipo "pedido de productos." Para crear los modelos sintáctico-semánticos y léxicos, comentados en las Secciones 2.1.3 y 2.1.4, hemos usados el corpus de 900 diálogos inicial, obteniendo los conjuntos $\alpha = \{MSSL_{Ti}\}$ y $\beta = \{ML_{Ti}\}$, i = 1 ... 43, pues el sistema Saplen genera 43 tipos distintos de prompts.

Para determinar el valor óptimo del umbral de similitud *t* (discutido en la Sección 2.2.1) hemos realizado experimentos considerando diversos valores de dicho umbral. Usando el simulador de usuarios, y empleado *EscenariosB*, hemos generado un corpus de 300 diálogos por cada valor de *t*, usando en todos los casos la técnica propuesta. El análisis de estos corpora de diálogos muestra que los mejores resultados se obtienen cuando *t* = 0,5.

Empleando el valor óptimo de *t*, se ha vuelto a usar el simulador de usuarios y *EscenariosA* para generar otro corpus de 900 diálogos. La Tabla 2 muestra los resultados medios (en %) obtenidos a partir de la evaluación de dicho corpus.

WA	SU	IR	TC
84,62	71,25	13,20	68,32

Tabla 2. Resultados de evaluación usando el sistema de RAH *mejorado*

4. CONCLUSIONES Y TRABAJO FUTURO

Comparando las Tablas 1 y 2 se observa que la técnica propuesta permite mejorar las tasas de exactitud de palabras (WA), comprensión de habla (SU), recuperación implícita (IR) y logro de tareas (TC) en 8,5%, 16,54%, 4% y 43,81% absoluto, respectivamente. Una línea de trabajo futuro es aplicar la técnica propuesta en otros sistemas de diálogo que empleen otros tipos de frases. Otra línea consiste en estudiar métodos alternativos para determinar el valor de confianza de las palabras empleadas en las correcciones. Por ejemplo, un método alternativo podría consistir en considerar un valor proporcional al número de fonemas en común entre la palabra errónea y la palabra que se usa para realizar la corrección.

5. BIBLIOGRAFÍA

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

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

[3] M. Jeong, B. Kim, G. G. Lee, "Semantic-oriented correction for spoken query processing", Proc. ICSLP, pp. 897-900, 1996

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

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

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

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

[8] M. Danieli, E. Gerbino, "Metrics for evaluating dialogue strategies in a spoken dialogue system", Proc. AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation, pp. 34-39, 1995