Faz
revista de diseño de interacción

Análisis de la Semántica Latente (LSA) y estimación automática de las intenciones del usuario en diálogos de telefonía (call routing)

Guillermo Jorge-botana, Ricardo Olmos**, José A. León**

Abstract

Latent Semantic Analysis and their multiple way in which is carried out, has been shown as an effecient technique to model the adquisition of human knoweleges and also various applications whitin the field of linguistic industry. One of its virtues is the classification of texts into previously established categories. This type of semantic categorisation can be used to try and identify the intentions of users when designing virtual agents within the telecoms industry. In this article we reflect on the significance of LSA within the design and implementation of virtual agents and we reflect on certain experiences which have had positive results in the past.

Resumen

El Análisis de la Semántica Latente (Latent Semantic Analysis) se ha mostrado como una técnica eficiente tanto para modelar la adquisición y representación del conocimiento humano como para diversas aplica-

* Departamento de Procesos Cognitivos, Facultad de Psicología (Universidad Complutense de Madrid).
** Departamento de Procesos Cognitivos, Facultad de Psicología (Universidad Autónoma de Madrid).
contacto: jorgeybotana@psi.ucm.es

ciones en el ámbito de la industria lingüística. Una de sus virtudes es la clasificación de textos en unas categorías semánticas previamente establecidas. Este tipo de categorización semántica puede ser implementada para tratar de identificar las intenciones de los usuarios en el diseño de agentes virtuales en el mundo de la telefonía. Aunque no contiene un experimento explícito, en este artículo se reflexiona sobre el lugar que ocupa LSA en el diseño e implementación de agentes virtuales y se lleva a cabo una revisión crítica de algunas experiencias previas que han tenido buenos resultados.

Palabras clave: Análisis de la semántica latente (LSA), Latent Semantic Analysis, Enrutamiento de llamadas (Call Routing), Gestión de Diálogos, Semántica, Agentes Virtuales, Análisis del Discurso, Modelos Probabilísticos del Lenguaje, Procesamiento del Lenguaje Natural (PLN).

1. Reconocimiento, enrutamiento y diálogos

Tanto los servicios telefónicos como las aplicaciones IVR/VRU (Interactive Voice Response) en general han introducido a las nuevas tecnologías ciertos parámetros que le son propios, en lo que respecta a la gestión y explotación automática de los diálogos que se producen en su ámbito. Según Dybkjær y Bernsen (2001), son tres los pilares que sustentan esta tecnología.

(1) El primero y más maduro es el que se refiere al propio reconocimiento de voz (IVR) y se basa principalmente en técnicas de reconocimiento de patrones de la señal de entrada. Estas técnicas han sido implementadas en herramientas que hoy en día son fáciles de encontrar en el mercado y que son las que están siendo empleadas masivamente en los servicios ofrecidos mediante telefonía móvil y fija. En estas aplicaciones, habitualmente se definen unas posibles entradas por medio de las combinaciones definidas en unas gramáticas[1]. El proceso empareja la entrada con uno de estos ejemplares definidos en las gramáticas bajo unos umbrales de probabilidad.

(2) El segundo es el que se refiere a la Gestión del Diálogo (Dialog Management) y trata de determinar qué hacer o por dónde encauzar al llamante en el supuesto de que haya sido reconocida una respuesta u otra. Este segundo proceso presupone un cierto conocimiento de las intenciones del usuario. Habitualmente se lleva a cabo por medio de preguntas dicotómicas y menús. Reconocida una entrada, se lanza una salida concreta a la aplicación principal y esta, mediante una estructura condicional, dirige la aplicación a uno u otro sitio. Las técnicas basadas en el Análisis de la Semántica Latente (de aquí en adelante LSA) también se pueden considerar dentro de este segundo pilar, es decir, dentro de la Gestión del Diálogo ya que LSA, aunque de otra manera, contribuye a decidir, a la vista de una entrada, cuál es la intención del usuario y dónde dirigirlo.

(3) En el tercero, Generación de las Salidas (Output Generation), se diseñan las respuestas que serán ofrecidas al usuario en el supuesto de una cierta demanda. Es importante introducir en su diseño modelos y directrices de usabilidad que provienen de la observación de las interacciones entre usuarios y agentes reales que fueron recopilados en las mismas situaciones que en las que se van a encontrar los agentes virtuales. Un sistema formal de evaluación de usabilidad en los diálogos lo proporciona CODIAL (Dybkjær, Bernsen

1 Una gramática es una formalización de la combinatoria de las posibles palabras o frases que puede responder un usuario y los valores que se devolverán en caso de detectar unas respuestas u otras. Además, se podrán asignar pesos a cada estructura según se estime su frecuencia de aparición. Póngase por caso que el sistema nos pregunta "Nuestro servicio le ofrece una amplia gama de equipos informáticos como ordenadores portátiles, de sobremesa o enrutadores, ¿en que tipo de equipo está interesado?. En este caso, se han de crear gramáticas para que el sistema IVR pueda reconocer por ejemplo estructuras del tipo "estoy interesado en portátiles", "estoy interesado en ordenadores de sobremesa", "en portátiles", "en ordenadores portátiles", "en routers", "quiero un portátil" y así hasta que cubra un segmento suficiente de las posibles entradas. Existen varios estándares entre los que se encuentran GSL, ABNF y grXML.

y Dybkjær, 1998), el cual, por un medio de códigos y descripciones, nos permite etiquetar los diseños en cuanto a sus posibles errores de usabilidad. Este sistema se puede obtener directamente de la página del proyecto DISC (http://www.disc2.dk/tools/codial/index.html) en el que se encuentran pautas y ejemplos de su uso. Es muy ventajoso emplear este tipo de estándares en el prototipado de la aplicación y pueden ser incluso integrados en los diseños VISIO y de otras aplicaciones en forma de códigos numéricos.

Como no pasa desapercibido, en estos dos últimos pilares (Gestión del Diálogo y Generación de las Salidas) entra en juego el conocimiento de las intenciones y deseos del usuario y la gestión de todo lo concerniente a sus percepciones , creencias, niveles de ansiedad ante una operación, sobreestimación de las recompensas, etc. En primer lugar se ha de conocer cuáles son los deseos del usuario y en segundo lugar, diseñar diálogos o bien para informar de los pasos para conseguirlos y los compromisos que contraerá con ello o bien para informar de la imposibilidad de llevarlos a cabo y cuál es la situación que se mantiene. Las técnicas basadas en LSA tiene el propósito de identificar y dar significado a los deseos de los usuarios de manera que puedan ser dirigidos a donde puedan llevarse a cabo. Dada una demanda, los sistemas basados en LSA la clasificarán en una categoría y dirigirán el flujo donde se informe o se actúe en torno a los tópicos de esa categoría.

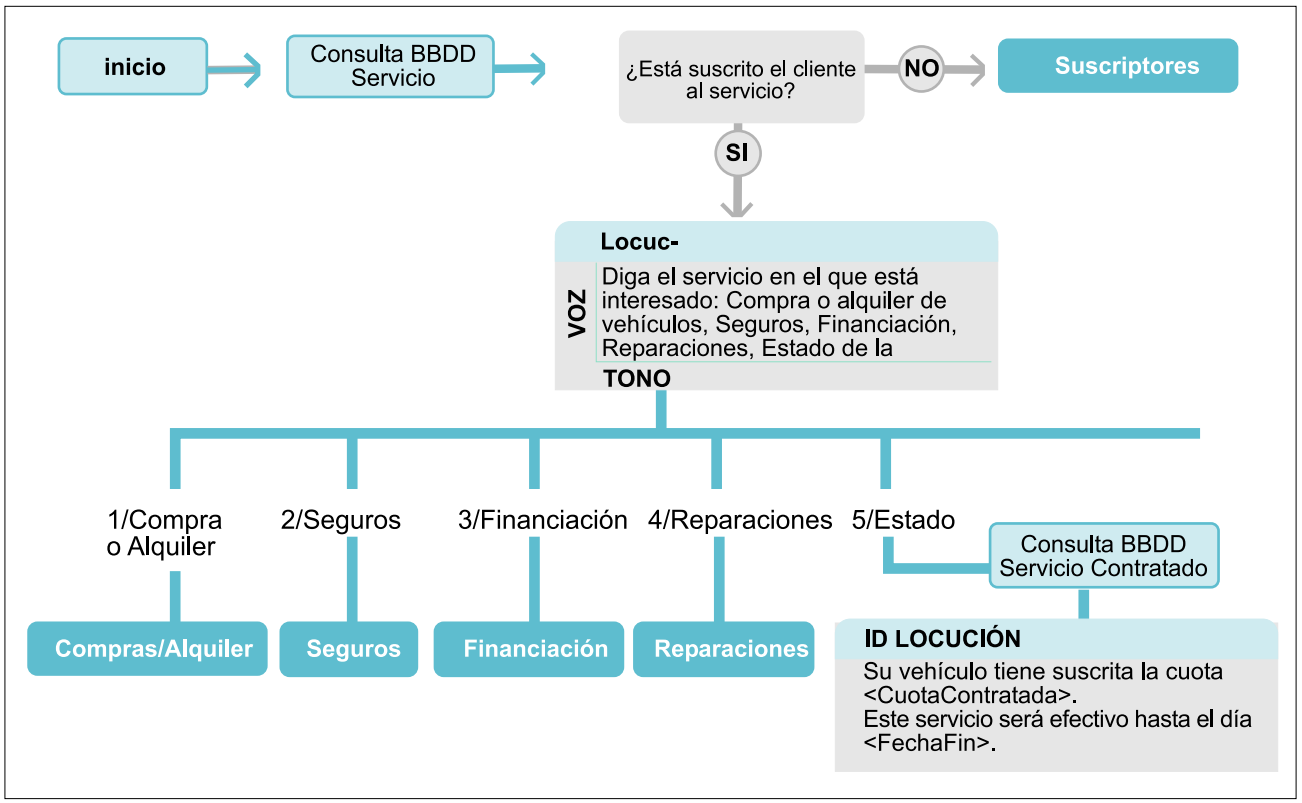
2. LSA y enrutamiento

La forma clásica de gestionar los diálogos y acciones en las aplicaciones de Gestión de Diálogo es el empleo de menús y preguntas más o menos cerradas. De esta manera, se puede acotar el rango de posibles respuestas para que estén contempladas en las gramáticas que la aplicación maneja, es decir, formen parte de las posibles combinaciones que las gramáticas ofrecen en esa parte de la aplicación. En otras palabras, para comparar las palabras o frases del usuario con los candidatos que sirven para enrutarle por una u otra rama de la aplicación. Una de las principales limitaciones de la forma clásica es que el usuario está obligado a verbalizar ciertos términos o expresiones previamente propuestos y es obligado, a su vez, a recorrer parte de la aplicación para ir perfilando lo que quiere. Así, sería imprescindible atravesar algunos menús antes de que el sistema esté en condiciones de ofrecer aquello que se viene buscando, lo que quizás sería un hándicap para los usuarios, en especial para los considerados expertos[2].

En la figura 1 se representa un ejemplo ficticio del diseño de la interacción entre sistema y usuario y con él podemos hacernos una idea de un diseño de estas características. Si buscásemos por ejemplo alquilar un vehículo, tendríamos que atravesar por lo menos dos menús. El primero sería el referente a la ruta Compras-Alquiler y seguramente, en lo sucesivo, atravesaríamos otro que identificase si se trata de una compra o de un alquiler. A su vez, por ejemplo, podríamos encontrar aún otra opción sobre si el tipo de vehículo a alquilar es un vehículo industrial, un vehículo de uso individual o vehículos de dos ruedas. Estas desventajas se dan, si cabe en mayor medida, para sistemas cuyos menús están regidos por tonos.

No obstante, algunas aproximaciones interesantes han tratado también de atajar el problema de la secuencialidad y abrir la posibilidad de que el usuario pueda saltarse o cambiar el orden de menús seleccionando directamente opciones que están presentes en niveles más profundos de la arquitectura de la aplicación. Destaca el uso de los Formularios de Iniciativa Mixta (Mixed Initiative Forms) que pueden ser implementados con estándar VXML (Voice eXtensible Markup Language version 1.0 W3C Note 05 May 2000) o la Interpretación Robusta del Lenguaje natural (Robust Natural

2 Si bien es verdad que la personalización de la aplicación a partir de la identificación del tipo de usuario (experto o no-experto) y la posibilidad de interrumpir las locuciones antes de que estas terminen puede mitigar la incomodidad generada por los continuos menús, esta no desaparecerá ya que aunque con menús abreviados y sin el acompañamiento de explicaciones redundantes, la navegación seguirá siendo secuencial y jerárquica.



Language Interpretation) propuesta en la plataforma NUANCE (NUANCE, 2001, p196). Esto permite que el usuario pueda tener la opción de demandar una información o acción que en ese momento no es ofrecida explícitamente y saltar literalmente a otra parte de la aplicación donde sí es ofrecida. Además, también se pueden proporcionar varias porciones de la información requerida en un mismo momento y en el orden que se desee[3]. Este último efecto es parecido al que se obtiene cuando se introducen datos de búsqueda en la caja de texto de "Google Maps". Se pueden introducir los datos en el orden que se quiera e incluso obviando algunos de ellos (http://maps.google.es/maps). Al igual que se reducen en "Maps" las cajas de texto, también se reducen en este tipo de aplicaciones el número de diálogos que requieren verbalización de datos.

Con todo, esta forma de implementación tiene varios inconvenientes, a saber, se hace inoperante cuando la variabilidad de los valores que pueden tomar las entradas es extremadamente grande (haciendo que una gran parte de estas entradas queden fuera del alcance de las gramáticas) y además, estas gramáticas están diseñadas para tener en cuenta el reconocimiento de los nexos sintácticos y el orden de las palabras lo cual agrava más la dificultad en un medio de extrema variabilidad.

Una forma de solventar parte de estos problemas son las técnicas basadas en modelos estocásticos del significado o modelos semánticos del lenguaje. Estas técnicas no son tan sensibles a las degradaciones léxicas y sintácticas pues no suelen conceder importancia al reconocimiento de palabras cerradas ni al orden y ocurrencia de las abiertas[4] y debido a la representación vectorial de un vocabulario muy amplio, pueden someter a categorización semántica estructuras de gran variabilidad. Existen herramientas cerradas para integrar modelos semánticos en aplicaciones de

[3] Merece la pena revisar http://www.developer.com/voice/article.php/3413361 para hacerse una idea de la filosofía de este tipo de diseños y como el usuario puede ofrecer la información requerida variando el orden inicial de la aplicación u ofreciéndola toda junta.

[4] Se apela a la distinción clásica entre palabras cerradas o de función y palabras abiertas o de contenido. Las primeras, determinantes, preposiciones, etc, carecen de significado y sirven como nexo entre las segundas: verbos, adjetivos, adverbios, etc.

gestión de diálogo como, por ejemplo, la aplicación "Call Steering" de NUANCE (http://www.nuance.com/callsteering/) o "Natural Language" de INFINITY (http://www.naturalanguage.es/). Sin embargo, este artículo se ocupará de analizar las experiencias llevadas a cabo con LSA. Como técnica basada en un modelo semántico del lenguaje, LSA tiene las mismas ventajas de los paquetes anteriormente mencionados[5], pero con la particularidad de que se puede controlar todo el proceso de modelado del lenguaje. Al igual que las demás aproximaciones de "Call routing", implementar un sistema con LSA significaría que no sería necesario proponer posibles entradas al usuario (posibles palabras a verbalizar), sino que simplemente, se le ofrecen preguntas iniciales y abiertas del tipo: "Bienvenidos a nuestros servicios bancarios, ¿en qué podemos ayudarle?". A partir de esta pregunta, la respuesta del usuario será identificada entre unas posibles categorías las cuales, condicionarán las rutas a seguir.

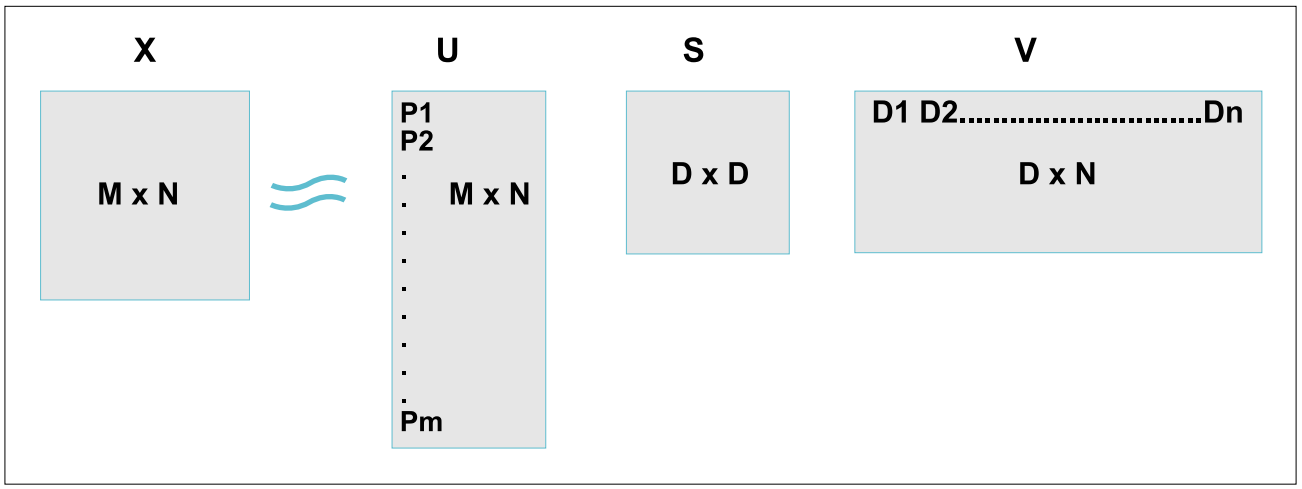
3. LSA como técnica de categorización de documentos

El Análisis de la Semántica Latente (LSA/LSI) fue originalmente descrito por Deerwester, Dumais, Furnas, Landauer y Harshman (1990) cómo un método de Recuperación de la Información (Information Retrieval). Fueron más tarde Landauer y Dumais (1996; 1997) los que propusieron este modelo cómo un modelo plausible de la adquisición y la representación del conocimiento. Desde ese momento ha sido empleada para modelar algunos fenómenos cognitivos (Landauer, 1999; Kintsch,1998; Kintsch, 2001; Kintsch y Bowles, 2002), además de aplicaciones más directas como son la corrección de textos en el ámbito académico (Trusso, 2005), para medidas de cohesión y coherencia textual (Graesser ,McNamara, Louwerse and Cai, 2004), para simular modelos de usuarios potenciales en usabilidad WEB (Blackmon, Polson, Kitajima, y Lewis, 2002; Blackmon y Mandalia, 2004; Jorge-Botana, 2006a,2006b) o como complemento a las ontologías (Cederberg y Widdows, 2003).

Para llevar a cabo la técnica, se procesa un texto de grandes dimensiones, lo que se conoce como el corpus lingüístico. El corpus se representa en una matriz cuyas filas contiene todos los términos distintos del corpus (palabras) y las columnas representen una ventana contextual en la que aparecen esos términos (habitualmente párrafos) (ver figura 2). De este modo, la matriz contiene sencillamente el número de veces que cada término aparece en un documento. Esta matriz sufre una ponderación que resta importancia a las palabras excesivamente frecuentes y la aumenta a las palabras moderadamente infrecuentes (Nakov, Popova, Mateev, 2001) con la idea de que las palabras demasiado frecuentes no sirven para discriminar bien la información importante del párrafo y las moderadamente infrecuentes sí. El siguiente paso es someter esta matriz ponderada a un algoritmo llamado Descomposición del Valor Singular (SVD), variante del análisis factorial (figura 2). El SVD se aplica con la idea de reducir el número de dimensiones de la matriz original en un número mucho más manejable (en torno a 300), sin que se pierda la información sustancial de la matriz original (Landauer y Dumais, 1997; Olde, Franceschetti, Karnavat, Graesser, 2002; Kurby, Wiemer-Hastings, Ganduri, Magliano, Millis, y McNamara, 2003). No obstante, el número de dimensiones depende de la naturaleza del corpus, por lo que puede ser muy variable y depender de varios criterios (Wild, Stahl, Stermsek y Neumann,2005)

Lo interesante de esta reducción de dimensiones no es únicamente mejorar el manejo de una matriz tan grande como la original, sino crear un espacio semántico vectorial en el que tanto términos como documentos están representados por medio de vectores que contengan sólo la información sustancial para la formación de conceptos (figura 3).

[5] Hágase la salvedad en cuanto a que los paquetes cerrados facilitan el reconocimiento de producciones espontáneas y su trascripción en base a las mismas muestras del lenguaje que servirán para la formación de los modelos semánticos. Entiéndase que LSA no es una herramienta ni un sistema sino un tipo de técnica en la que se basa la arquitectura de algunos sistemas.



:: Figura 2. Desglose de la matriz principal en las dos matrices de vectores singulares y una matriz diagonal de valores singulares. Será a partir de este desglose desde donde se reducirán las dimensiones tomando sólo las que más capacidad tienen para diferenciar regiones semánticas.

La nueva representación de los términos y documentos en este espacio semántico ha mostrado ser muy exitosa simulando comportamientos humanos. La ventaja de representar el lenguaje vectorialmente es que éstos son susceptibles de comparaciones por medio de cosenos, distancias euclídeas u otras medida de similitud (figura 4). Además, a partir de las coordenadas de los términos ya representados pueden introducirse en el espacio nuevos vectores que representen textos producidos a posteriori y que se suelen llamar pseudodocumentos (Landauer, Foltz y Laham, 1998). Será a partir de estos pseudodocumentos como se lleve a cabo la categorización de textos, transcripciones y diálogos[6]. En la medida en que se obtengan cosenos altos entre dos pseudocumentos, se podrá inferir que ambos versan sobre una temática similar. En nuestro caso, tendremos un vector con la verbalización del usuario y otro con el texto que represente una categoría en la línea de negocio de modo que cuando sean similares (ver figura 4), se considerará que se refieren a los mismos contenidos y se dirigirá al usuario a los diálogos y acciones pertinentes.

4. Algunos casos concretos de LSA y CALL ROUTING

Una de las primeras experiencias en el uso de LSA en servicios de telefonía fue llevado a cabo en los laboratorios de Lucent Technologies por Chu-Carroll y Carpenter (1999). Toman como corpus de referencia 4.497 transcripciones telefónicas en las que los clientes interactuaban con los operadores de un "Call Center" de un servicio bancario. Analizan primero las características de las trascripciones y en especial las primeras producciones verbales del cliente, de las cuales hacen una taxonomía según sean los datos aportados por él (1. - Nombre del destino - ej: "alguien en leasing, por favor"; 2. – Actividad -ej:"Querría hablar con alguien sobre cuentas de ahorro"; 3. - Demanda indirecta dónde se dan rodeos - ej: "Un amigo me dijo que si llamaba y había comprado un coche como él…"). Además, consignan dónde enrutan los operadores a los clientes dadas esas primeras demandas o producciones verbales. Siguiendo con su análisis del corpus encuentran que el 20% de las llamadas necesitan más información para

[6] Para familiarizarse con la técnica e incluso hacer algunas pruebas, puede visitarse el sitio LSA que mantiene, aunque algo desactualizado, la universidad de Boulder http://lsa.colorado.edu/ . También merece la pena echar un vistazo a http://www.cs.utk.edu/~lsi/. Además, en nuestro grupo de interés hemos creado un sitio en el que exponemos alguna documentación y se pueden realizar pruebas sobre algunos espacios semánticos específicos de dominio http://www.elsemantico.com/.

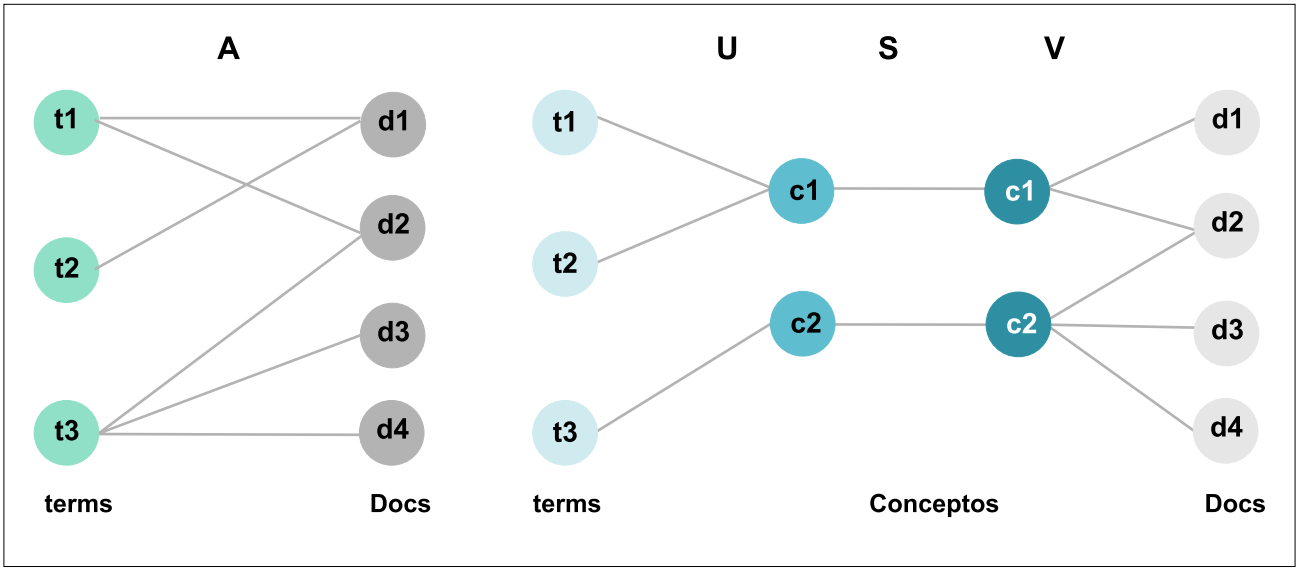


Figura 3: Representación esquemática de lo que significa la reducción de dimensiones llevada a cabo por medio de SVD. En la figura de la izquierda, cada término está representado por cuatro dimensiones, tantas como párrafos existen en el corpus.{d1,d2,d3,d4}. En la figura de la derecha, los términos pasan a estar representados por dos dimensiones abstractas pero de una mayor utilidad funcional. A cada término se le infiere una probabilidad de estar representado en un concepto. Compruébese por ejemplo que al término t2 se le infiere cierta probabilidad de salir en el párrafo d2 aunque como muestra la figura de la izquierda, esto no se produzca.

ser desambiguadas y no son dirigidas directamente. De estas llamadas que requieren de más información, un 75% se producen por la falta de especificación en los nombres de las frases. Por ejemplo: "crédito para coche" sin especificar si se trata de un crédito de un coche que ya existe o de un crédito para uno nuevo. El otro 25% se producen por la poca especificación de los verbos en las frases como por ejemplo "deposito directo" sin la especificación de si lo que quiere es abrir un depósito o cambiar uno existente. Basado en esta falta de especificidad, introducen un módulo posterior al que contiene LSA que vuelva a requerir al usuario más información para desambiguar su demanda y de nuevo categorizar la reformulación de la demanda con el sistema LSA.

En resumen, Chu-Carroll y Carpenter proponen un sistema en que ante la respuesta del usuario a una pregunta abierta del tipo "Diga algo"(Say Anything), el módulo de LSA categorice dicha producción y proponga algunos candidatos de enrutación (si sólo hay uno se enruta directamente y si no hay ninguno se le pasa con un agente). Si es el caso que hubiese diversos candidatos, el siguiente módulo, el de desambiguación, dada una respuesta de usuario concreta, formulará una pregunta para que sea el mismo usuario el que reformule su demanda. En la figura 5 se puede ver la arquitectura de dicho sistema.

La forma de entrenar el módulo LSA es la siguiente: como cada demanda del usuario está marcada con el destino final donde fue dirigido, todas las frases que fueron dirigidas a un destino forman un documento único. De 3.753 llamadas se formarán 23 destinos que representarán cada uno un documento[7]. Antes de haberse formado estos documentos, se han suprimido del corpus la lista de palabras no deseadas como lo son las palabras más comunes o llamadas listas-stop (stop list) y las palabras que forman parte del ruido introducido en el lenguaje espontáneo (fillers) cuya lista se llama "lista a ignorar" (ignore list). Una particularidad del tratamiento del texto es la que viene dada por la formación de bigramas y trigramas. Los bigramas y trigramas son términos que por su uso conjunto forman una unidad. Ejemplos serían "car+loan", "check+acount+balance", etc. Los auto-

[7] La técnica LSA parte de una matriz términos-documentos en la que se consignan las ocurrencias de los primeros en los segundos.

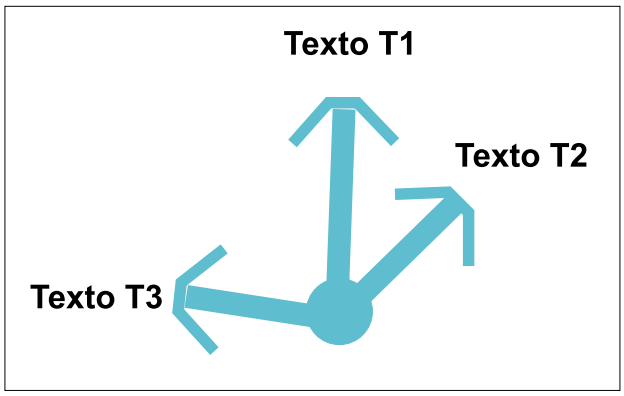


Figura 4: El resultado final del proceso es un espacio vectorial en el que están representados palabras y documentos y al que se le pueden integrar documentos nuevos. Como se puede ver en la figura, cuando se compara la similitud semántica entre tres textos dentro del espacio semántico definido por LSA, tenemos que los textos 1 y 2 son parecidos porque forman un ángulo cerrado y por lo tanto su coseno es próximo a 1. La relación semántica de los textos 1 y 2 con el tercero es casi nula. De esta manera, dos textos o dos palabras son susceptibles de comparación en base a medidas operativas lo que permite describir las relaciones de significado.

res introducen una lista de estas palabras (si su ocurrencia en conjunción es significativa) para encontrar en el corpus la ocurrencia de estos términos y unirlos de manera que salgan en el corpus como un término indiferenciado. La forma de hacerlo es buscar las ocurrencias de los unigramas , bigramas y trigramas pero de manera que si es encontrada una de orden mayor como un trigrama, también son conservadas las de orden menor. Si por ejemplo, es encontrada "check+acount+balance", también serán introducidos en el corpus final los términos "check+acount", "check", "acount" y "balance". De esta manera se preservan las apariciones de los órdenes menores. Además, como habitualmente se hace, se calcula el corrector IDF[8]. Calculados todos los pasos de LSA incluido SVD y la reducción de dimensiones, obtenemos las matrices de consulta[9].

Obtenidas estas matrices de consulta y dada una demanda del usuario, es calculada su representación vectorial como pseudodocumento y es establecida la comparación con cada uno de los documentos destino. Los documentos con alta similitud bajo un umbral serán los candidatos que se introducirán en el módulo de desambiguación. Hay que resaltar que para el cálculo de la similitud entre destino y pseudodocumento emplean como medida los cosenos pero corregidos con el propósito de maximizar las diferencias. Para ello, emplean la función sigmoidea extraída de la distribución de las similitudes (medidas con el coseno) entre cada llamada y cada destino, dividido entre 1 si fue enrutado a ese destino o 0 si no lo fue. Dada esta distribución, se ajusta a una función logística y se obtiene la función concreta (índice de confianza) que modula los cosenos crudos.

$$Conf(d_a, d_b, x) = 1/(1 + e^{-(d_a x + d_b)})$$

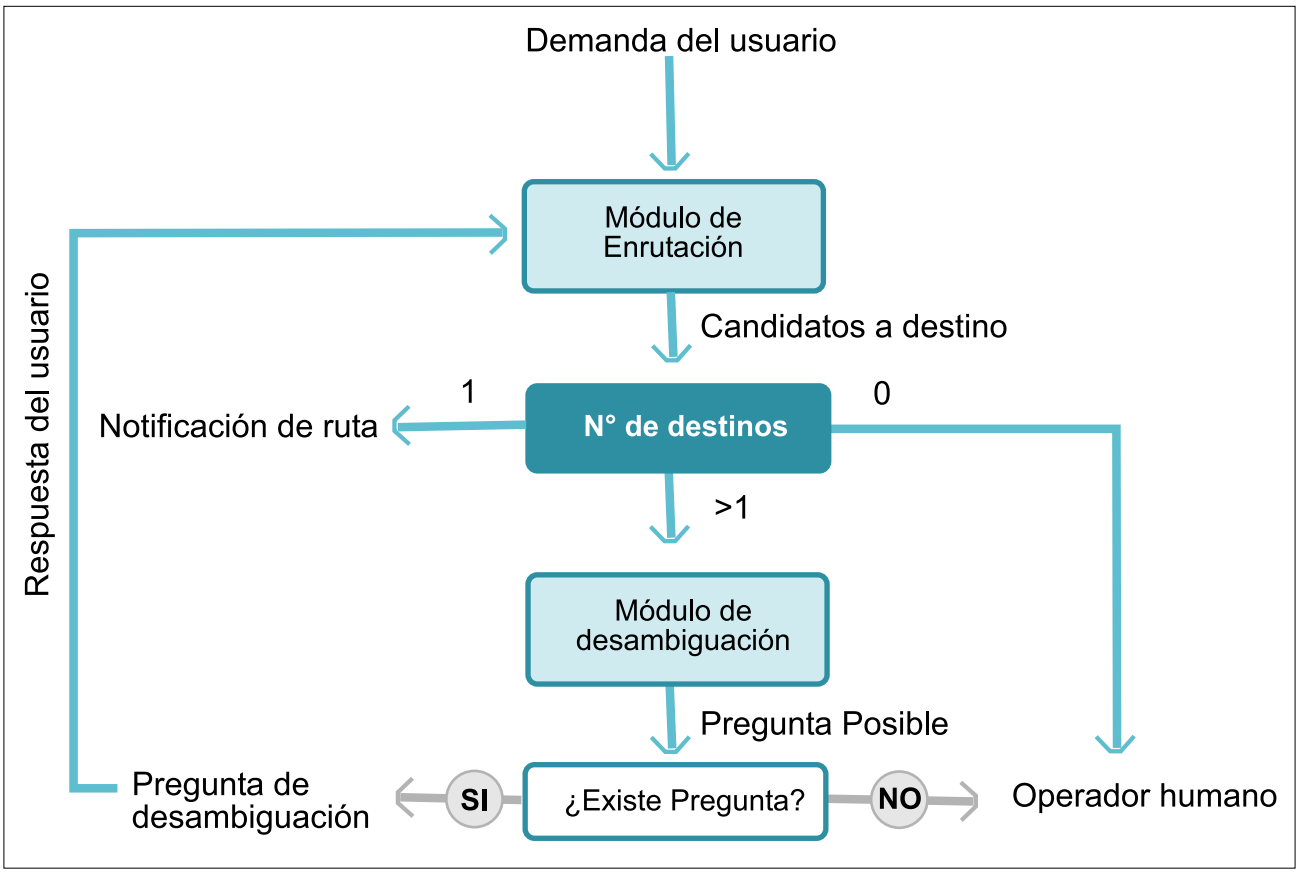
Donde **X** es el coseno entre la demanda concreta del usuario y un destino concreto.
.**d**a y **d**b son coeficientes de la función sigmoidea para el vector destino.

Empleando la función sigmoidea se obtiene una reducción del error del 16.7% de llamadas bien dirigidas. Obtenidos los índices de similitud se calcula empíricamente un umbral sobre el cual se considera que el destino es un posible candidato. Como resultado se obtiene que 0.2 representa el mejor umbral para el índice de confianza.

Si más de un candidato supera dicho umbral se pone en marcha el módulo de desambiguación. El módulo de desambiguación intenta que dados dos destinos posibles, el usuario reformule la demanda para que uno de esos dos destinos se descarte. Para ello, hace uso de la filosofía de LSA en cuanto a la representación vectorial. Si el vector demanda está muy parejo a dos vectores destino, se habrán de encontrar tér-

[8] IDF es uno de los métodos usados en la llamada fase de preproceso antes de someter la matriz a SVD. La motivación de este método es ponderar cada término en base a su importancia para representar supuestos dominios semánticos. Se infiere que si un término ocurre en un número muy alto de documentos será mal predictor del dominio al que puede pertenecer. Imagínese el lector un término como "dolor" en un corpus basado en una taxonomía médica. Este término no nos ofrecería gran información sobre tipos de enfermedades, no así por ejemplo "inmunodeficiencia". IDF trata de menguar el influjo de términos muy frecuentes y poco informativos como "dolor".

[9] Matrices de consulta: son las matrices que contienen la representación vectorial de términos y documentos.



:: *Figura 4. Tomado de Carroll y Carpenter(1999).*

minos que representen a los vectores-diferencia del vector-demanda con cada uno de los vectores-destino. De esta forma se comprueba qué términos son los que pueden ajustarse a esos vectores diferencia. Una vez encontrados estos, sólo servirán como candidatos los que pueden formar un n-grama con la demenda original. Es decir, para una demanda como "loans please", y extrayéndose dos posibles destinos: "loan services" y "costumer lending", se han de extraer los términos que se ajustan a los vectores diferencia entre "loans please" -"loan services" y "loan please" -"costumer lending" respectivamente. Dada la lista de términos que se ajustan a los vectores diferencia, serán sólo relevantes para la construcción de la nueva pregunta aquellos que pueden formar un n-grama con "loan". De esto resultarán términos como "auto-loan" y "loan-payoff". De aquí se crearán preguntas más o menos estándar a modo de moldes. En este ejemplo, para crear la pregunta, dado que todas comparten "loan", la pregunta propuesta será "What kind of loan?". De esta manera, el usuario reformulará la demanda, volverá a pasar por el módulo de enrutamiento y si es el caso de que hay más de dos destinos, volverá también al módulo de desambiguación y sufrirá el mismo proceso hasta ser refinada del todo. Si en el módulo de desambiguación, se pudiese formar sólo un n-grama como "exist+car+loan", entonces se propondría una pregunta del tipo "sí-no" como "Is this about an exiting car loan'". Este sistema consigue un porcentaje de aciertos del 93.8% e incluso tiene buenos resultados teniendo en cuenta los errores propios del reconocimiento del habla (97%-92.5% en su máximo rendimiento y 75%-72% en su rendimiento más modesto).

A partir de este anterior trabajo, los autores Cox y Shahshani (2001) hacen un análisis de la conveniencia o no de emplear LSA para "call routing". En primer lugar critican la forma en que Chu-Carroll y Carpenter (1999) forman los documentos. Cox y

Shahshani (2001) proponen dos formas de crear los documentos en LSA y "call routing". La primera a la que ellos llaman T-Route y que es la empleada por Chu-Carroll y Carpenter (1999), se basa en que un documento está formado por todas las llamadas que fueron dirigidas a un mismo sitio. Formando los documentos de esta manera, existirán tantas columnas o documentos como rutas posibles haya en nuestra lógica de negocio. La otra manera a la que llaman T-Trans se basa en que cada trascripción por separado configure un documento. De esta manera, habrá tantos documentos como llamadas. El problema de aplicar T-Route es que el número de columnas o documentos en (M,N) es muy pequeño lo que hace que los términos queden representados con una dimensionalidad que puede ser muy baja (la dimensionalidad de los términos no podrá superar al número de columnas). Recordemos que la elección de N dimensiones para representar términos y documentos es arbitraria y viene dada por la propia técnica de Descomposición del Valor Singular (SVD). Otro problema de T-Route es el siguiente: el número de términos de un documento consulta o demanda suele ser muy reducido por lo que al formarse surgirá un vector (antes de introducirlo en el espacio vectorial) con la mayoría de sus índices ocupados por ceros. Esto contrasta con los documentos con los que se quieren comparar, los cuales, surgen de la compilación artificial de todas las llamadas que se enrutaron a un destino en un solo documento, lo que provoca que haya abundancia de valores no-cero e incluso valores abultados que muestran una sobre-representación. Tomando medidas de porcentajes de error en varios espacios semánticos formados de diferentes formas (con T-Route o con T-Trans, Sin dimensionalizar, dimensionalizado con LSA o Análisis discriminante despues de LSA), Cox y Shahshani (2001) encuentran que T-Trans tiene en todo momento menores porcentajes de error que T-Route. También encuentra que los espacios no dimensionalizados tienen mejores rendimientos que cuando se reducen las dimensiones, excepción hecha de la combinación que permite mejores resultados: dimensionar con LSA a 350 dimensiones y posteriormente empleando 31 dimensiones en el análisis discriminante. Estos resultados muestran de alguna forma la aseveración de Cox y Shahshani (2001), los cuales dicen que en "call routing", el escenario varía frente a otras utilidades de LSA. Para "Call Routing", el número de dimensiones viene dado a priori y es igual al número de rutas posibles por lo que puede ser innecesario descubrir cual es la dimensionalidad óptima con LSA. Esto parece confirmarse con el buen comportamiento de los espacios sin dimensionar aunque puede deberse, tal como nosotros mismo hemos comprobado en Olmos, León, Escudero, Jorge-Botana (en prensa, 2007), a que no exista suficiente variabilidad en los corpus de referencia, es decir, que no estén representados en el corpus términos que representan información tangencial, lo cual no invalidaría la técnica LSA para corpus de diálogos telefónicos de mayor cobertura. Como concluyen Franceschetti, Karnavat, Marineau, McCallie, Olde, Terry, Graesser (2001), los corpus en los que se conserva esta información tangencial, funcionan mejor a la hora introducir comparaciones por medio de pseudodocumentos. En cualquier caso, el experimento de Cox y Shahshani (2001), muestra que aunque todos los espacios tienen un comportamiento aceptable (16%-6% de error), existen formas de LSA en combinación con otras técnicas que pueden resultar muy beneficiosas.

Otra extensión interesante de LSA para clasificación de diálogos es la propuesta de Serafín y Dí´Eugenio (2004). En su experiencia emplean FLSA (Featured Latent Semantic Analysis) para llevar a cabo clasificaciones de diálogos. Para ello prueban el comportamiento de tres corpus marcados previamente. CallHome, un corpus de llamadas telefónicas en español, MapTask que contiene diálogos en torno a las instrucciones en torno a un mapa y Diag-NLP que versa sobre diálogos sobre el aprendizaje del uso de ordenadores. Todos estos corpus están marcados con etiquetas que aluden a varios criterios. El método FLSA computa dos matrices que luego concatena: la matriz términos documentos y la matriz etiquetas-documentos. La matriz etiquetas-Documentos está formada por las etiquetas de los propios corpus e identifica a cada documento. Esta matriz resultante[10], (w+t)*D es tratada de la misma forma que se trataría en la

[10] (w+t)*D se refiere a la concatenación de la matriz de palabras por documentos WxD y la matriz de etiquetas documentos TxD.

	Ventajas	Desventajas
Chu-Carroll y Carpenter (1999)	Módulo de desambiguación en base a n-gramas. Corrección de los cosenos en base a valores empíricos.	- Formación de documentos: cada destino en la línea de negocio se considera un documento. Cada uno de los documentos lo forman todas las llamadas que fueron enrutadas a ese destino. Hay un reducido número de documentos los cuales resultan abultados
Cox y Shahshani (2001)	- Argumentación de la conveniencia o no de la reducción de dimensiones en el caso de enrutación de llamadas (call routing). - Buenos resultados en combinatoria con otras técnicas estadísticas.	- En la mayoría de los casos no encuentra mejores resultados con la reducción de dimensiones lo que puede deberse a varias causas (ver texto).
Cox y Shahshani (2001) Serafín y Di'Eugenio (2004)	- Integración de las etiquetas en el análisis. (FLSA o Featured Latent Semantic Análisis). - FLSA obtiene mejores resultados que LSA y que las líneas base. - También obtienen ventaja frente a resultados obtenidos anteriormente.	

forma clásica de LSA. De alguna manera, las etiquetas son tratadas como términos ocupando también filas en la matriz de datos. De esta forma se consigue forjar más cohesión entre los propios documentos y términos en coalición con las etiquetas artificiales. Ambos casos LSA y FLSA se comportan de una forma muy efectiva a la hora de categorizar diálogos pero los resultados muestran que FLSA se comporta algo mejor.

5. Conclusiones

Se han presentado en este artículo algunas observaciones en torno al lugar que pueden ocupar las técnicas basadas en LSA en el diseño y desarrollo de agentes virtuales. LSA puede clasificar las demandas de los usuarios de un servicio telefónico en las categorías que identifican cada una de las líneas de negocio. En otras palabras, LSA emitirá un juicio de cuán parecido es el texto-demanda del usuario con cada uno de los documentos que representan las posibles rutas. Por tanto, las técnicas basadas en LSA pueden ser implementadas como herramientas en los módulos de Gestión del Diálogo. Esta parte de la aplicación empieza una vez se ha producido el reconocimiento del habla espontánea. Dada una entrada del usuario, LSA se encargará de clasificarlo en alguna de las categorías semánticas. Las técnicas basadas en espacios vectoriales tienen algunas ventajas sobre las técnicas clásicas de menús y tonos, a saber, dada la posibilidad de valorar entradas de discurso libre, ofrecen al usuario la flexibilidad de no seguir unas pautas marcadas exclusivamente por el sistema, evitando atravesar un excesivo número de menús . Otra ventaja es que toda interacción parte de una pregunta inicial del tipo "Diga algo" ("Say Anything"). De esta manera, el usuario percibe más naturalidad en los diálogos que mantiene. En el caso de querer reconocer la respuesta a una pregunta abierta, implementar tal sistema definiendo gramáticas, conllevaría un riesgo para el propio sistema de reconocimiento de voz o un excesivo número de items en

los menús. Aún así, para llevar a cabo un sistema de enrutamiento basado en LSA es preciso conocer que tipo de materia prima estamos empleando por lo que serían de mucha utilidad más estudios sobre el tipo de corpus, su preproceso y la manera de representarlo en la matriz de ocurrencias como variables que inciden en la efectividad del enrutamiento. Por ejemplo, una cuestión importante es la que sugieren los resultados de Cox y Shahshani (2001) y es la relativa a si en todos los corpus o bajo todos los preprocesos y tratamientos es beneficiosa la reducción de dimensiones o si por el contrario, hay ocasiones en que no es necesaria e incluso contraproducente. Por otro lado, el futuro de este tipo de técnicas pasa ahora por implementar algoritmos que empleen como base el espacio semántico que emana de LSA y que extraigan el sentido de estructuras concretas insertas en los textos. Un ejemplo de esto es el algoritmo de predicación (Kintsch, 2001) donde trata de encontrar el sentido a estructuras predicativas y metafóricas.

En nuestro grupo de interés, estamos trabajando tanto en encontrar las propiedades de los corpus que elevan la eficiencia de LSA (Olmos et al, en prensa, 2007; León, Olmos, Escudero, Cañas, Salmeron, 2006) como los parámetros involucrados en la representación de estructuras basadas en predicaciones (Jorge-Botana, Olmos, León, Molinero, 2007). En definitiva, el reto actual se encuentra en desarrollar parseadores que localicen cierto tipo de estructuras y aplicar sobre ellos algoritmos que operen sobre el espacio semántico resultante del proceso LSA.

Bibliografía

Blackmon, M.H.; Mandalia, D.(2004). Steps of the cognitive Walkthrough for the web(CWW):. Navigation System Analysis. Institute of Cognitive Science. University of Colorado.Boulder.

Blackmon, M.H., Polson, P.G., Kitajima, M. & Lewis, C. (2002). Cognitive Walkthrough for the Web.In CHI 2002: Procedings of the conference on Human Factors in Computing Systems,463-470.

Cederberg, S. y Widdows D.(2003). Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction.Human Language Technology Conference archive. Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4 table of contents. Edmonton, Canada

Chu-Carroll,J., y Carpenter,B.,(1999) Vector-based natural language call routing, Computational Linguistics, v.25 n.3, p.361-388,

Cox, S. And Shahshahani, B. (2001). A Comparison of some Different Techniques for Vector Based Call-Routing. Proc. 7th European Conf. on Speech Communication and Technology, Aalborg.

Debra Trusso Haley, Pete Thomas, Anne De Roeck, Marian PetreA Research (2005) Taxonomy for Latent Semantic Analysis-Based Educational Applications. Technical Report of Open University - Department of Computing.

Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A. (1990). Indexing by latent semantic analysis, Journal of the Society for Information Science, 41(6), 391-407."

Dybkjær, L, Bernsen, N.O.,(2001) Usability evaluation in spoken language dialogue systems. Annual Meeting of the ACL archive. Proceedings of the workshop on Evaluation for Language and Dialogue Systems - Volume 9, Toulouse, France

Dybkjær, L., Bernsen, N.O., and Dybkjær, H.(1998): A methodology for diagnostic evaluation of spoken human-machine dialogue. International Journal of Human Computer Studies (special issue on Miscommunication), 48, 1998, 605-625.

Franceschetti, D.R., Karnavat, A., Marineau, J., McCallie, G.L., Olde, B.A., Terry, B.L., & Graesser, A.C. (2001). Development of physics test corpora for latent semantic analysis. Proceedings of the 23th Annual Meeting of the Cognitive Science Society (pp. 297-300). Mahwah, NJ: Erlbaum.

Graesser, A., McNamara, D.S., Louwerse, M., & Cai, Z. (2004). Coh-Metrix: Analysis of text on cohesion and language. Behavioral Research Methods, Instruments, and Computers, 36, 193-202.

Jorge-Botana, G., Olmos, R., León J.A. Molinero, P., Variantes a la extracción automática de vecinos semánticos con LSA y al algoritmo de predicación (Kintsch, 2001). Informe técnico. http://www.elsemantico.com/Documentos/verdaderosentido.pdf

Jorge-Botana, G(2006a) Adecuación de ruta: nuevo índice basado en el Análisis de la Semántica Latente. No Solo Usabilidad journal, nº 5. 3 de Mayo de 2006. http://www.nosolousabilidad.com/articulos/adecuacion_ruta.htm

Jorge-Botana, G(2006b) El Análisis de la Semántica Latente y su aportación a los estudios de Usabilidad . No Solo Usabilidad journal, nº 5. 5 de Enero de 2006.http://www.nosolousabilidad.com/articulos/analisis_semantica_latente.htm

Kintsch, W. and Bowles, A. (2002) Metaphor comprehension: What makes a metaphor difficult to understand? Metaphor and Symbol, 2002, 17, 249-262"

Kintsch, W.(2001) Predication. Cognitive Science 25, 173-202

Kintsch,W.(1998).The Representation of Knowledge in Minds and Machines.International Journal of Psychology.Volume 33, Number 6 / December 1, 1998 pp:411 - 420

Kurby, C. A., Wiemer-Hastings, K., Ganduri, N., Magliano, J. P., Millis, K. K., & McNamara, D. S. (2003). Computerizing reading training: Evaluation of a latent semantic analysis space for science text. Behavior Research Methods, Instruments, & Computers, 35, 244-250

Landauer , T. K., (1999). Latent semantic Analysis is a Theory of the Psychology of Language and Mind. Discourse Processes, 27, 303-310.

Landauer, T.K., & Dumais, S.T. (1997). A solution to PlatoÕs problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review."

Landauer, T.K., Foltz, P.W., Laham, D. (1998). An introduction to latent semantic analysis. Discourse Processes, 25, 259-284."

Leon, J.A., Olmos, R., Escudero, I., Cañas, J.J. & Salmeron, L. (2006). Assessing short summaries with human judgments procedure and latent semantic analysis in narrative and expository texts. Behavior Research Methods, Instruments, and Computers.

Nakov,P.I. , Popova,A.., and Mateev, P.(2001) Weight Functions Impact on LSA Performance, (Sofia University Press, Sofia, 2001).

NUANCE Speech recognition system, version 8.0: grammar developer´s guide, 2001. Nuance communication, Inc. http://community.voxeo.com/vxml/docs/nuance20/grammar.pdf

Olde, B. A., Franceschetti, D.R., Karnavat, Graesser, A. C. & the Tutoring Research Group (2002). The right stuff: Do you need to sanitize your corpus when using latent semantic analysis? Proceedings of the 24th Annual Conference of the Cognitive Science Society (pp.708-713). Mahwah, NJ: Erlbaum."

Olmos, R, León J.A., Escudero, I, Jorge-Botana, G.. El papel de los corpus en la evaluación de resúmenes con análisis semántico latente (LSA) Revista Signos (en prensa).

Serafin, R. And Di Eugenio. B., FLSA: Extending Latent Semantic Analysis with features for dialogue act classification. ACL04, 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July.
http://www.cs.uic.edu/~bdieugen/PS-papers/acl04.pdf

Voice eXtensible Markup Language version 1.0 W3C Note 05 May 2000.

Wild. F, Stahl. C , Stermsek. G, Neumann. G : Parameters Driving Effectiveness of Automated Essay Scoring with LSA, in: Proceedings of the 9th International Computer Assisted Assessment Conference (CAA), 485-494, Loughborough, UK, July, 2005.

Guillermo de Jorge Botana, es Licenciado en psicología por la Universidad Complutense de Madrid y Magíster en Psicolingüística Aplicada. Presentó su tesina sobre los modelos computacionales en el acceso al léxico escrito y actualmente prepara su tesis sobre la técnica de Análisis de la Semántica Latente (LSA) como modelo informático de la comprensión del texto y del discurso: una aproximación distribuida al análisis semántico. Profesionalmente ha trabajado en el mundo de las tecnologías tanto en su fase de desarrollo y programación como en la fase de análisis y diseño. Actualmente trabaja en INDRA programando y diseñando aplicaciones para plataformas IVR.

Ricardo Olmos, es Licenciado en Psicología y actualmente trabaja como consultor en SPSS Ibérica. Además, es doctorando en la Universidad Autónoma de Madrid y prepara su tesis en torno al Análisis Semántico Latente: "El Análisis Semántico Latente (LSA), ¿es una teoría psicológica o únicamente una herramienta de análisis semántico?".

José A. León, es profesor titular en el Departamento de Psicología Básica en la UAM. Sus temas de interés se han centrado en el estudio de los procesos cognitivos que intervienen en la comprensión del lenguaje, en la cognición causal y sus aplicaciones en la semántica y en la adquisición de conocimiento. Cuenta con más de un centenar de publicaciones nacionales e internacionales.