

# Hacia el Uso de la Información Sintáctica y Semántica en los Sistemas de Búsqueda de Respuestas

Diego Mollá

Centre for Language Technology  
Division of Information and Communication Sciences  
New South Wales 2109  
Australia  
diego@ics.mq.edu.au

**Resumen:** Los sistemas actuales de búsqueda de respuestas por lo general combinan métodos basados en la recuperación de la información y la extracción de la información. Dichos métodos se aprovechan de la rapidez de los algoritmos resultantes y la presencia de redundancia en la información, pero pocos de ellos usan métodos basados en información lingüística más allá del nivel de las palabras. En este artículo presentamos un estudio del impacto de la información sintáctica y semántica en el proceso de selección de la sentencia candidato final. Dicho proceso de selección se basa en medidas de solapado de relaciones gramaticales y solapado de formas lógicas planas.

**Palabras clave:** formas lógicas, búsqueda de respuestas

**Abstract:** Current question answering systems typically combine methods based in information retrieval and information extraction. Such methods leverage the speed of the resulting algorithms and the presence of information redundancy, but few systems use methods based on linguistic information beyond the word level. In this paper we present a study of the impact of the syntactic and semantic information in the selection process of the final candidate sentence. The selection process is based on measures of grammatical relation overlap and flat logical form overlap.

**Keywords:** logical forms, question answering

## 1. *Introducción*

Recientemente hay un gran interés en el desarrollo de sistemas de búsqueda de respuestas capaces de procesar documentos de texto automáticamente y de encontrar la respuesta a preguntas arbitrarias en tiempo real. Tal interés es motivado por la disponibilidad de la información, el incremento en la potencia de ordenadores, y la disponibilidad de recursos y herramientas.

**Disponibilidad de la información.** La cantidad de textos disponibles digitalmente ha alcanzado enormes proporciones y sigue creciendo exponencialmente. Un ejemplo es la cantidad de documentos indexados por Google, que es de 4,285,199,774 (9 Marzo 2004). Cualquier empresa hoy en día dispone de documentos almacenados digitalmente, hasta tal punto que por lo general almacenan más información digitalmente que en forma impresa. Hay otras fuentes de información disponible, como por ejemplo enciclopedias digitales y manuales de uso de software. Toda

esta información, aunque disponible digitalmente, es difícil de buscar para encontrar la información exacta deseada. Los sistemas actuales de recuperación de información (como los motores de búsqueda) son útiles para tal efecto pero hay situaciones en que es necesario encontrar respuestas exactas en vez de documentos relevantes, sobre todo cuando el usuario dispone de poco tiempo para encontrar la información deseada en la posiblemente larga lista de documentos.

**Potencia de los ordenadores.** La potencia de los ordenadores sigue creciendo exponencialmente desde hace varias décadas. Cualquier ordenador personal a la venta tiene por lo general más potencia que los grandes sistemas de ordenadores de décadas atrás. En consecuencia es posible el intentar resolver aplicaciones que requieran ejecuciones más costosas, tanto en la dimensión temporal como la espacial.

**Disponibilidad de herramientas.** Actualmente hay disponibles varias herramien-

tas y recursos que facilitan la tarea de búsqueda de respuestas. Así, se pueden encontrar fácilmente analizadores sintácticos con gramáticas de gran cobertura, sobre todo para el inglés, como Link Grammar (Sleator y Temperley, 1993) y Conexor FDG (Järvinen y Tapanainen, 1997), paquetes estadísticos y herramientas de tratamiento de colecciones de textos, diccionarios electrónicos y otras herramientas léxicas como WordNet (Fellbaum, 1998), módulos de tratamiento de textos como separadores de palabras, etiquetadores de entidades, e incluso herramientas para la comprobación de argumentos lógicos como OTTER (Kalman, 2001). La disponibilidad de tales herramientas hace posible la construcción de sistemas de búsqueda de respuestas y otros sistemas complejos de tratamiento del lenguaje natural en un plazo de tiempo relativamente corto, y de forma asequible.

## 2. *Sistemas de Búsqueda de Respuestas*

Los sistemas de búsqueda de respuestas, sobre todo los participantes en la sección de búsqueda de respuestas de la Text REtrieval Conference TREC<sup>1</sup> (Voorhees, 2001), tienen que ser capaces de encontrar la respuesta buscando en una cantidad de texto relativamente grande (unos 3GB repartidos entre un millón de documentos aproximadamente para los sistemas de TREC). Para ello varios sistemas toman provecho de la inherente redundancia de información que se encuentra en tal cantidad de texto. De este modo algunos de los sistemas no intentan examinar detalladamente el texto sino que se conforman con buscar simples patrones de respuesta o contar la frecuencia de ciertas palabras clave. Dichos sistemas por lo general se centran en la *rapidez* de proceso y la asunción de que, si el sistema no detecta la respuesta en el documento *X*, tal vez la detecte en otro documento *Y* que expresa la respuesta de forma más obvia. Varios de dichos sistemas incluso incrementan la redundancia de la información por medio del uso de textos adicionales disponibles a través de la *World Wide Web*.

En una arquitectura típica como la de la Figura 1 los sistemas reducen progresivamente la cantidad de información a proce-

sar, y en cada etapa usan métodos más intensivos de análisis (Hirschman y Gaizauskas, 2001; Voorhees, 2001). Así, en una etapa previa a la sesión de preguntas y respuestas, un **indexador** analiza los documentos y extrae toda la información posible, generalmente las palabras clave y sus posiciones relativas. Las resultantes imágenes de los documentos se guardan en una base de datos que se usará durante la sesión de preguntas y respuestas. Durante la sesión de preguntas y respuestas, un **analizador de preguntas** clasifica la pregunta y determina el tipo de respuesta esperada. Por ejemplo, dicho módulo determina los tipos posibles de entidades que pueden responder a la pregunta. Así, es de esperar que la respuesta a una pregunta de la forma *Who ...?* sea una persona o una organización. El analizador de preguntas también devuelve un análisis de la pregunta, por lo general más detallado que la información que el indexador había producido de los documentos. Dicha información puede variar entre una simple lista de palabras (Cooper y Rüger, 2000, por ejemplo) o una forma lógica de cierta complejidad (Harabagiu y Maiorano, 2000; Elworthy, 2000, por ejemplo). El siguiente módulo es el **preseleccionador** que determina cuáles de los documentos pueden contener la respuesta. El preseleccionador compara la imagen de la pregunta con las imágenes de los documentos producidas por el indexador y por lo general usa técnicas de recuperación de la información como el modelo de espacios vectoriales o modelos booleanos. Dichos modelos suelen considerar los documentos como simples “bolsas de palabras” (*bag of words*) sin estructura ni dependencias (Voorhees, 2001). Los documentos resultantes pasan por un **filtro** que retiene las oraciones o pasajes que tengan más posibilidad de contener la respuesta. Aquí es cuando por lo general se recompensa a los fragmentos de texto que tengan entidades compatibles con el tipo de respuesta aceptable. Finalmente, un **valorizador** realiza un análisis más intensivo de los fragmentos supervivientes y les asigna un valor relacionado con la posibilidad de que la respuesta se encuentre en dicho texto. A tal efecto el valorizador usa toda la información de que pueda disponer, como las estructuras sintácticas de la pregunta y las oraciones que contienen la posible respuesta, recursos externos de información como ontologías y diccionarios electrónicos, o inclu-

---

<sup>1</sup>trec.nist.gov

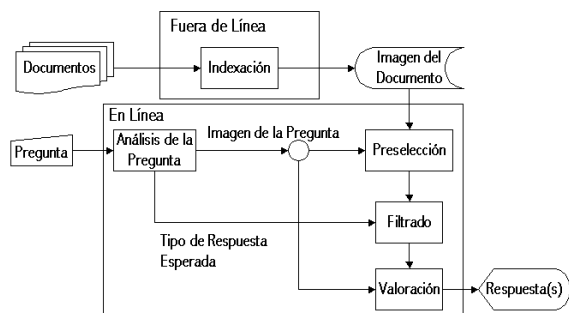


Figura 1: Sistema de búsqueda de respuestas genérico

so evaluadores de consistencia lógica entre la pregunta y la posible respuesta. Algunos sistemas (Harabagiu et al., 2001, por ejemplo) incorporan mecanismos de evaluación de la calidad de salida de cada etapa y sistemas de realimentación para conseguir la cantidad ideal de información a pasar entre las etapas. Otros sistemas (Soubbotin, 2001, por ejemplo) utilizan métodos de patrones de texto que pueden tener la respuestas para determinar la respuesta final. Dichos patrones pueden llegar a ser bastante complejos y pueden estar organizados jerárquicamente.

Aun así, el uso de la información sintáctica y semántica en tales sistemas es la excepción en vez de la norma. En el presente estudio nos hemos concentrado en el valorizador, y hemos realizado varios experimentos en un intento de determinar hasta qué punto se puede usar la información sintáctica y semántica.

En la Sección 3 describimos brevemente el sistema que hemos usado. Pasamos a describir el uso de información sintáctica y semántica en las Secciones 4 y 5, respectivamente. Describimos la evaluación en la Sección 6, donde también mostramos y comentamos los resultados. Finalmente indicamos las conclusiones y líneas de investigación futura en la Sección 7.

### 3. El Sistema

Dado que nuestro interés es la evaluación del impacto de diferentes niveles de información lingüística en el proceso de búsqueda de respuestas, el sistema en sí se basa en una simplificación de la arquitectura presentada en la Sección 2. Así, el sistema utiliza los documentos facilitados por NIST<sup>2</sup>, que son el resultado de pasar las preguntas por un sistema de búsqueda de respuestas. De este modo

evitamos el tener que implementar el indexador y el preseleccionador y podemos centrarnos en la tarea de búsqueda de respuestas en sí. El analizador de preguntas utiliza un total de 29 patrones regulares y el conjunto final de tipos de preguntas está basado en el tipo de respuesta esperada: *person, date, location, money, number, city, date, organization, percent, country, state, river, name, unknown*. Los patrones están basados en el conjunto de 500 preguntas de TREC 2002, y la precisión del clasificador para ese conjunto es del 78.6%. Una prueba con las preguntas de TREC 2003 devolvió una precisión del 77% (incluyendo las preguntas clasificadas correctamente como *unknown*), lo que prueba que los patrones generalizan bien ante preguntas desconocidas.

Para simplificar, el sistema no devuelve la respuesta exacta sino la oración que contiene la respuesta, truncada a 250 caracteres. El resultante sistema por lo consiguiente es apto para tareas de búsqueda de respuesta como las definidas en la sección de búsqueda de respuestas de las conferencias TREC 1999 hasta TREC 2001 y la tarea de pasajes de TREC 2003, en la que una versión de este sistema sin el analizador de preguntas y sin el extractor de entidades participó, dando resultados ligeramente superiores a la mediana (Mollá, 2003).

El filtro se encarga de seleccionar las oraciones con más posibilidad de tener la respuesta. En un primer paso, el filtro divide los documentos en oraciones por medio de una expresión regular sencilla que identifica varios caracteres de puntuación como delimitadores de final de oración. Después, el filtro usa el extractor de entidades de GATE<sup>3</sup> para obtener las entidades que aparecen en el candidato. Dado que el extractor de GATE usa un número reducido de entidades, el clasificador de preguntas utiliza la conversión del Cuadro 1 para determinar el tipo final de la respuesta esperada. Finalmente, el filtro combina la información del tipo de respuesta esperada con el solapado de palabras entre la pregunta y el candidato de respuesta. Así, cada candidato adquiere un valor que es la suma del número de palabras en común (sin contar palabras repetidas ni palabras de parada como artículos, puntuación, etc.<sup>4</sup>) con la

<sup>3</sup><http://gate.ac.uk>

<sup>4</sup>Las palabras de parada fueron obtenidas de <http://www-fog.bio.unipd.it/waishelp/>

<sup>2</sup>[www.nist.gov](http://www.nist.gov)

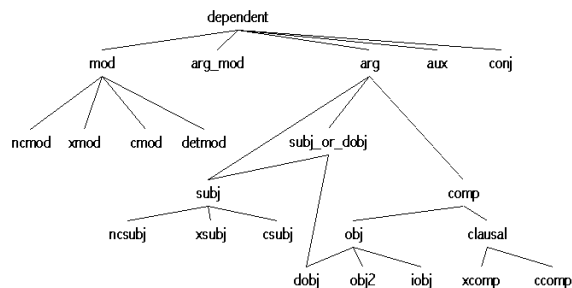


Figura 2: Jerarquía de relaciones gramaticales (Briscoe y Carroll, 2000)

respuesta, más 10 puntos si el candidato tiene una entidad del mismo tipo que la respuesta esperada.

Por último, el valorizador identifica la oración que contiene la respuesta. Para ello el valorizador usa la información sintáctica y semántica que describimos en las secciones siguientes. Este es el único módulo que varía en nuestras evaluaciones. De este modo aseguramos que la única variable a tener en cuenta en las evaluaciones corresponde a lo que queremos evaluar: el impacto de la información sintáctica y semántica en las tareas de búsqueda de respuestas.

#### 4. Relaciones Gramaticales

Con la intención de poder evaluar analizadores sintácticos que tengan salidas potencialmente incompatibles, Carroll, Briscoe, y Sanfilippo (1998) diseñaron un conjunto de relaciones gramaticales para usar a modo de lingua franca. Las relaciones gramaticales desarrolladas en tal estudio presentan una clasificación jerárquica (Figura 2). De este modo es posible representar la salida de analizadores sintácticos que produzcan árboles sintácticos de diferente granularidad. Ese es el motivo de relaciones gramaticales como `subj` y `dobj` para representar el sujeto y el objeto directo, respectivamente, si el analizador sintáctico puede diferenciar entre estos conceptos, o relaciones más generales como `subj_or_dobj` en caso contrario. El Cuadro 2 muestra las relaciones gramaticales que aparecen en este artículo. Una explicación más detallada de todas las relaciones gramaticales aparece en (Briscoe y Carroll, 2000).

Por ejemplo, las relaciones gramaticales de la oración *The man that came ate bananas and apples with a fork without asking son:*

```

DETMOD(.,man,the),
CMOD(that,man,come),
SUBJ(.,man,come),
SUBJ(.,man,eat),
DOBJ(.,banana,eat),
DOBJ(.,apple,eat),
CONJ(.,apple,banana,and),
NCMOD(.,eat,with,fork),
DETMOD(.,a,fork),
XCOMP(.,ask,without,eat)

```

Es muy fácil comprobar la similitud entre dos oraciones por medio de las relaciones gramaticales. Aunque en principio hay que tener cuidado de comprobar si una relación gramatical en una oración subsume a otra en la otra oración (`SUBJ_OR_DOBJ(eat,man)` subsume `SUBJ(eat,man,.)`, por ejemplo), en la práctica, dado que el mismo analizador sintáctico se ha usado en ambas oraciones, simplemente tenemos que calcular la intersección entre los conjuntos de relaciones gramaticales, lo que llamamos el *solapado de relaciones gramaticales* (*srg*). En otras palabras, cada relación gramatical es una “palabra” que no tiene estructura a la hora de hacer la comparación.

El generador de relaciones gramaticales que usamos en el presente estudio toma como datos de entrada el resultado del analizador sintáctico “Conexor Functional Dependency Grammar” (Tapanainen y Järvinen, 1997). Los valores de recuperación y precisión (59.7 % y 74.6 %, respectivamente) según el conjunto de oraciones usado por Carroll, Briscoe, y Sanfilippo (1998) indican que el proceso puede mejorarse.

#### 5. Formas Lógicas Planas

En un intento de incorporar más información semántica, hemos estudiado el uso de formas lógicas planas como las que se usan en (Harabagiu et al., 2001; Lin y Pantel, 2001; Mollá et al., 2000). En particular hemos adoptado las formas lógicas de (Mollá et al., 2000). Dichas formas lógicas usan *reificación* como instrumento para aplanar expresiones con estructuras anidadas. Así, la forma lógica de *The cp command will quickly copy files* es:

```

object('cp',o2,[x2]),
object('command',o3,[x3]),
compound_noun(x2,x3),
prop('quickly',p5,[e6]),
evt('copy',e6,[x3,x7]),
object('file',o7,[x7])

```

<i>Tipo de Pregunta</i>	<i>Respuesta Esperada</i>
country, city, state, river	location
percent	number
name	person O organization O location
Otros tipos de pregunta devuelven el mismo tipo de respuesta	

Cuadro 1: Conversión entre los tipos de preguntas y las respuestas esperadas

<i>Relación</i>	<i>Descripción</i>
CONJ(tipo,cabeza+)	Conjunción
MOD(tipo,cabeza,dependiente)	Modificador
CMOD(tipo,cabeza,dependiente)	Modificador clausal
NCMOD(tipo,cabeza,dependiente)	Modificador no clausal
DETMOD(tipo,cabeza,dependiente)	Artículo
SUBJ(cabeza,dependiente,rg_inicial)	Sujeto
OBJ(cabeza,dependiente,rg_inicial)	Objeto
DOBJ(cabeza,dependiente,rg_inicial)	Objeto directo
XCOMP(cabeza,dependiente)	Complemento clausal sin sujeto

Cuadro 2: Relaciones gramaticales que aparecen en el presente artículo

Es decir, hay dos entidades,  $x_2$  y  $x_3$ , que representan los objetos de los sustantivos *cp* y *command*, respectivamente. Hay otra entidad,  $x_7$  (un fichero), y otra entidad  $e_6$  que representa el evento del comando  $x_3$  copiando el fichero  $x_7$ . El evento  $e_6$  ocurre rápidamente.

Del mismo modo que las relaciones gramaticales, las formas lógicas planas no son más que conjuntos de predicados. Para encontrar la similitud entre dos oraciones no tenemos más que calcular la intersección entre los conjuntos de predicados de ambas oraciones, lo que llamamos el *solapado de formas lógicas* (*sfl*). A diferencia de las relaciones gramaticales, no obstante, esta vez tenemos que analizar la estructura de los predicados, pues las variables usadas para referir a las entidades ( $x_1$ ,  $x_2$ , etc) pueden variar. Véase, por ejemplo, la forma lógica de la pregunta *Which command copies files?*:

```
object('command', o1, [x1]),
evt('copy', e2, [x1, x2]),
object('file', o2, [x2])
```

Si queremos comparar esta forma lógica con la de *The cp command will quickly copy files* arriba, tenemos que comprobar si, siempre que aparece la variable  $o_1$  en un predicado de la pregunta, aparece la variable  $o_3$  en el mismo argumento en la respuesta, y así con el resto. Estas operaciones son muy fáciles

de hacer si convertimos las variables de la pregunta en auténticas variables de Prolog y usamos el mecanismo de unificación inherente de Prolog. El Cuadro 3 muestra ejemplos simples de formas lógicas para dos preguntas similares, una posible respuesta, y la intersección.

Como hay varias maneras de establecer correspondencias entre variables, la correspondencia que nos interesa es la que produce el conjunto intersección con mayor número de elementos.

El generador de formas lógicas utiliza el resultado del analizador sintáctico “Conexor Functional Dependency Grammar” (Tapanainen y Järvinen, 1997) a través de un proceso “bottom-up” descrito en (Mollá y Hutchinson, 2002).

## 6. Evaluación y Resultados

Para comprobar el efecto de los métodos de valoración de respuestas usamos los conjuntos de preguntas de TREC 2003 y los patrones proporcionados por Ken Litkowsky y que aparecen en las páginas de Web de TREC.<sup>5</sup> Estos patrones han sido hechos a mano tras inspeccionar todas las ocurrencias de respuestas correctas en todos los sistemas participantes. Los patrones por lo tanto no cubren todas las formas posibles de respuestas correctas, y siempre es posible que

<sup>5</sup><http://trec.nist.gov>

Pregunta	Forma Lógica Plana
<i>Did John see Mary?</i>	<b>object('mary',O,[X]), evt('see',E,[Y,X]), object('john',O2,[Y])</b>
<i>Did Mary see John?</i>	<b>object('john',O,[X]), evt('see',E,[Y,X]), object('mary',O2,[Y])</b>
Candidato	Forma Lógica Plana
<i>John saw Mary</i>	object('john',o1,[x1]), object('mary',o3,[x3]), evt('see',e2,[x1,x3])

Cuadro 3: Búsqueda de respuestas con formas lógicas planas. Intersección de formas lógicas en negrita.

un patrón dé como correcta a una respuesta que no tiene nada que ver con la pregunta, por ejemplo si la respuesta es una fecha o un número. Por lo tanto, los resultados que presentamos abajo tienen que aceptarse de forma relativa. No indican la precisión exacta del sistema, pero pueden dar una indicación muy clara del impacto relativo de los métodos de valoración usados.

Los métodos de valoración son:

**Solapado de palabras (*sp*).** Tras varios experimentos comprobamos que la solución óptima consiste en usar una lista de palabras de parada, y no contar palabras repetidas.

**Solapado de relaciones gramaticales (*srg*).**

El método descrito en la Sección 4.

**Solapado de formas lógicas (*sfl*).** El

método descrito en la Sección 5.

El Cuadro 4 muestra los resultados de usar varios métodos de valoración de respuestas.

Método	Precisión
<i>sp</i>	14.77 %
<i>srg</i>	17.43 %
<i>sfl</i>	19.37 %

Cuadro 4: Experimentos con datos de TREC 2003.

Los resultados indican una mejora de la precisión en el solapado de relaciones gramaticales y, sobre todo, en el solapado de formas lógicas. Para comprobar si los métodos se complementan a la hora de escoger la respuesta hemos experimentado con unas simples combinaciones lineales de los tres métodos. Los resultados preliminares (Cuadro 5) no parecen indicar ninguna manera obvia de combinar los métodos para mejorar los resultados.

La Figura 3 muestra el incremento de la precisión (medida como el número de res-

Fórmula	Precisión
$3sfl + sp$	16.70 %
$3srg + sp$	15.25 %
$9sfl + 3srg + sp$	19.61 %
$9srg + 3sfl + sp$	18.16 %
$3srg + sfl$	17.91 %
$3sfl + srg$	19.13 %

Cuadro 5: Combinaciones de solapados y resultados con las preguntas de TREC 2003.

puestas correctas) si aceptamos que la respuesta válida esté dentro de las primeras  $x$  oraciones devueltas, donde  $x$  es el parámetro del eje horizontal en el gráfico de la figura. Los resultados indican una clara desventaja del solapado de palabras, mientras que el solapado de relaciones gramaticales y el solapado de formas lógicas toman valores parecidos.

El Cuadro 6 muestra el porcentaje de las respuestas correctas según el tipo de pregunta para las preguntas de TREC 2003. Para obtener resultados más objetivos la clasificación fue hecha manualmente. Aparte de algunas fluctuaciones menores no hay ningún tipo de pregunta que dé resultados diferentes en proporción a los resultados generales, excepto por el tipo *country* (“país”). Actualmente estamos explorando las causas de la diferencia en este tipo de preguntas. Nótese que la categoría con más miembros es *unknown* (“desconocida”), lo que indica la necesidad de refinar las clases de preguntas para acomodar más tipos.

## 7. Sumario y Conclusiones

Hemos presentado un estudio del uso de la información sintáctica y semántica en sistemas de búsqueda de respuestas por medio de medidas de solapado. Para ello hemos experimentado con el solapado de relaciones gramaticales y el solapado de formas lógicas planas. Nuestros resultados muestran una mejora con la introducción del solapado de relaciones gramaticales y especialmente con el so-

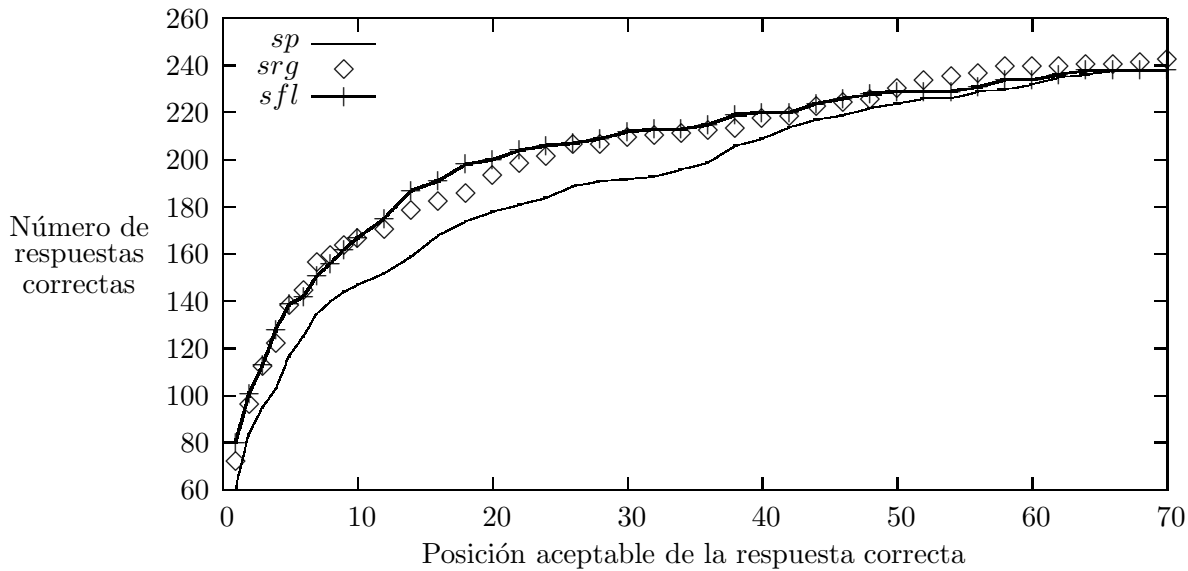


Figura 3: Incremento de la precisión con respecto a la tolerancia en la posición de la respuesta correcta

Tipo	N	sp	srg	sfl
City	18	17%	28%	33%
Name	17	24%	18%	18%
Country	23	17%	35%	48%
Percent	1	100%	100%	100%
Number	104	17%	13%	15%
Person	33	15%	12%	21%
State	4	0%	0%	0%
Location	26	31%	38%	31%
Money	3	33%	33%	0%
Date	46	7%	11%	11%
Organization	17	18%	29%	24%
River	2	0%	50%	0%
Unknown	119	9%	13%	16%

Cuadro 6: Respuestas correctas por tipo de pregunta (preguntas de TREC 2003),  $N$  indica el número de preguntas

lapado de formas lógicas. Los resultados conforman con la hipótesis de que, a mayor información disponible, la precisión del sistema de búsqueda de respuestas mejora.

Nótese que, para simplificar la evaluación y garantizar un tratamiento igual de las medidas de solapado, hemos evitado el introducir técnicas adicionales para mejorar los resultados. Por ejemplo, no hemos implementado ningún tratamiento específico para los tipos de preguntas. Tampoco hemos incluido ningún componente que tome ventaja de las características del dominio, aparte de la rudimentaria clasificación de respuestas y del extractor de entidades de GATE. Teniendo en cuenta que, aún así, el resultado del sola-

pado de formas lógicas supera la mediana de los valores presentados en la tarea de pasajes de la sección de búsqueda de respuestas de TREC 2003 (18.2%), y teniendo en cuenta que nuestra evaluación automática suele generar valores más bajos que una evaluación manual como la del TREC 2003 debido a la existencia de pasajes correctos que no correspondan exactamente a los patrones predefinidos en la evaluación, nuestros resultados son más que satisfactorios.

Ahora que hemos confirmado el potencial del uso de las formas lógicas, estamos estudiando medidas que se aprovechen de la riqueza de la información disponible. Un posible modo de aprovechar tal tipo de información incluye el desarrollo de patrones de relaciones gramaticales y formas lógicas especializados a los tipos de preguntas, de modo parecido a como los sistemas de búsqueda de respuesta actuales usan patrones de expresiones superficiales. Otra posibilidad que estamos estudiando es el análisis automático de los fragmentos de formas lógicas resultantes del solapado para determinar la respuesta exacta a la pregunta. También estamos desarrollando un clasificador de preguntas más preciso, y estamos estudiando la manera de refinar la clasificación de las preguntas para reducir el número de preguntas sin clasificar. También estamos considerando la introducción de recursos léxicos como WordNet y el uso de herramientas lógicas como OTTER.

También estamos estudiando el uso de me-

didadas de solapamiento más complejas que incluyan la introducción de diferentes pesos a los diferentes tipos de predicados de la forma lógica según el tipo de respuesta esperada, y la introducción de métodos abductivos como los usados en (Harabagiu y Maiorano, 2000).

Siempre es posible que el tipo de preguntas y respuestas usados en TREC no favorezcan el uso de información lingüística más allá de palabras sueltas y sus derivados, con lo que también estamos estudiando tipos más complejos de preguntas que requieran la combinación de información de varias fuentes. Claramente se necesita más investigación en este área.

### **Bibliografía**

- Briscoe, Ted y John Carroll. 2000. Grammatical relation annotation. On-line document. <http://www.cogs.susx.ac.uk/lab/nlp/carroll/grdescription/index.html>.
- Carroll, John, Ted Briscoe, y Antonio Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. En *Proc. LREC98*.
- Cooper, Richard J. y Stefan M. Rüger. 2000. A simple question answering system. En Ellen M. Voorhees y Donna K. Harman, editores, *Proc. TREC-9*, número 500-249 en NIST Special Publication. NIST.
- Elworthy, David. 2000. Retrieval from captioned image databases using natural language processing. En *Proc. CIKM 2000*.
- Fellbaum, Christiane, editor. 1998. *WordNet: an electronic lexical database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.
- Harabagiu, Sanda, Dan Moldovan, Marius Paşca, Mihai Surdeanu, Rada Mihalcea, Roxana Gîrju, Vasile Rus, Finley Lăcătuşu, y Răzvan Bunescu. 2001. Answering complex, list and context questions with LCC's question-answering server. En Voorhees y Harman (Voorhees y Harman, 2001).
- Harabagiu, Sanda M. y Steven J. Maiorano. 2000. Acquisition of linguistic patterns for knowledge-based information extraction. En *Proc. LREC-2000*.
- Hirschman, Lynette y Rob Gaizauskas. 2001. Natural language question answering: The view from here. *Natural Language Engineering*, 7(4):275–300.
- Järvinen, Timo y Pasi Tapanainen. 1997. A dependency parser for english. Informe Técnico TR-1, Department of Linguistics, University of Helsinki, Helsinki.
- Kalman, John A. 2001. *Automated Reasoning with OTTER*. Rinton Press, Paramus, NJ.
- Lin, Dekang y Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360.
- Mollá, Diego. 2003. Answerfinder in trec 2003. En *Proc. TREC 2003*.
- Mollá, Diego, Rolf Schwitter, Michael Hess, y Rachel Fournier. 2000. Extrans, an answer extraction system. *Traitement Automatique des Langues*, 41(2):495–522.
- Mollá, Diego y Ben Hutchinson. 2002. Dependency-based semantic interpretation for answer extraction. En *Proc. 2002 Australasian NLP Workshop*.
- Sleator, Daniel D. y Davy Temperley. 1993. Parsing English with a link grammar. En *Proc. Third International Workshop on Parsing Technologies*, páginas 277–292.
- Soubbotin, M. M. 2001. Patterns of potential answer expression as clues to the right answers. En Voorhees y Harman (Voorhees y Harman, 2001).
- Tapanainen, Pasi y Timo Järvinen. 1997. A non-projective dependency parser. En *Procs. ANLP-97*. ACL.
- Voorhees, Ellen M. 2001. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378.
- Voorhees, Ellen M. y Donna K. Harman, editores. 2001. *The Tenth Text REtrieval Conference (TREC-10)*, número 500-250 en NIST Special Publication. NIST.