Extracción automática de respuestas para documentación técnica

Fabio Rinaldi y Elia Yuste
Instituto de Lingüística Computacional,
Universidad de Zúrich, Suiza.
Winterthurerstrasse 190, CH-8057 Zúrich,
Correo electrónico: {rinaldi, yuste}@ifi.unizh.ch

Resumen: Este artículo presenta un sistema de Extracción de Respuestas que opera mediante la transformación de documentos y preguntas en una representación semántica denominada Forma Lógica Mínima (FLM). Un tratamiento especial de la terminología permite adecuar el sistema a dominios técnicos.
Palabras clave: Extracción de Respuestas, Análisis semántico, Forma Lógica

Abstract: This paper presents an Answer Extraction system which works by transforming documents and queries into a semantic representation called Minimal Logical Form (MLF). A special treatment of terminology makes it particularly suited to technical domains.
Keywords: Answer Extraction, Question Answering, Semantic Analysis, Logical Form

1. Introducción

Tradicionalmente, se ha asumido que los Sistemas de Recuperación de Información (SRI) tienen que encontrar documentos de apoyo sobre un tema en concreto, sin ocuparse de localizar la información de interés contenida en tales documentos. Ésto ha dado lugar a que algunos autores hayan observado que la Recuperación de Información (RI) tradicional debería llamarse "Recuperación de Documentos" (Rinaldi et al., 2002b).

En la última década muchos investigadores han centrado su interés en el desarrollo de sistemas que no sólo localicen documentos relevantes, sino que sean capaces de señalar la unidad informativa exacta en la que el usuario esté interesado. La serie de conferencias celebradas bajo el título de *"Message Understanding Conferences"* ha supuesto un gran avance en este campo. El concepto de Extracción de Información (EI) ha ido evolucionando y redefiniéndose progresivamente hasta el punto que hoy ya se le considera un área de investigación en si misma. Tales sistemas acostumbran a extraer tipos de información específicos y predefinidos por los propios creadores de la aplicación. Un problema fundamental al que se tienen que enfrentar estas aplicaciones más complejas es que el sistema, al estar ajustado a las plantillas predefinidas, no puede adaptarse fácilmente a nuevas plantillas como se esperaría si se cambia de dominio o se pretende responder a las necesidades de otro tipo de usuarios.

La Extracción de Respuestas ("Answer Extraction")[1](Abney, Collins, y Singhal, 2000) es una tecnología más reciente que tiene como objetivo resolver algunas de las cuestiones previamente citadas. Normalmente, los sistemas de Extracción de Respuestas (ER), los cuales permiten que el usuario formule preguntas arbitrarias, tratan de recuperar un pequeño fragmento textual que proporcione una respuesta a partir de un corpus determinado. En los últimos años la sección de ER de las competiciones TREC (Voorhees, 2000) se ha encargado especialmente de fomentar la investigación en esta dirección.

En este artículo presentamos un sistema de Extracción de Respuestas (ExtrAns) (sección 2) y su aplicación práctica en dos dominios diferentes. Una vez proporcionados los detalles de los componentes de procesamiento sintáctico (sección 3) y semántico (sección 4), mostramos cómo se realiza la extracción de respuestas (sección 5) y establecemos una comparación con un sistema típico de RI (sección 6). Como colofón, se presentan los principales resultados junto con un breve resumen de trabajo relacionado con estos temas (sección 7).

[1]Comúnmente denominada también "Question Answering"

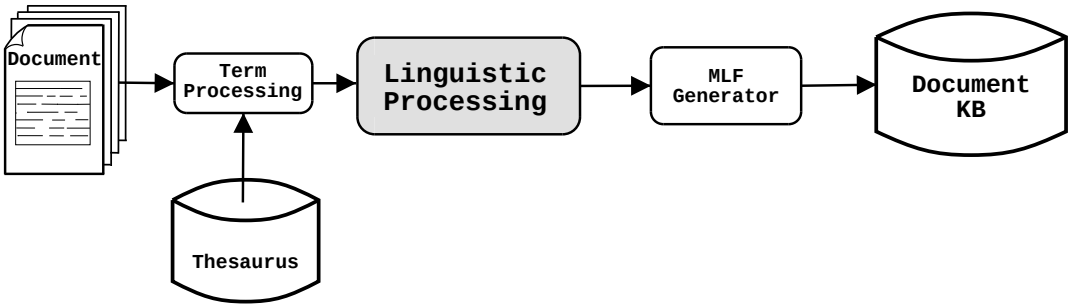

Figura 1: Arquitectura simplificada del sistema ExtrAns: procesamiento de los documentos

2. *La Arquitectura de ExtrAns*

ExtrAns es un sistema complejo que se compone de varios módulos distintos, escritos en diferentes lenguajes de programación. Cabe mencionar que el procesamiento tiene lugar en dos fases: se procesa una determinada colección de documentos fuera de línea (véase la figura 1), si bien la pregunta misma es procesada en línea (figura 2). Se lleva a cabo el mismo análisis lingüístico en ambas fases, transformando todo en una representación semántica llamada Forma Lógica Mínima (FLM). Los resultados de este análisis se almacenan en una Base de Conocimento (Rinaldi et al., 2003).

Esta arquitectura se ha testado con dos dominios técnicos diferentes. En un principio, las respuestas a preguntas arbitrarias de un usuario se extraían de los archivos de documentación de Unix (conocidos en inglés como *man pages*). El sistema abarca una serie de más de 500 páginas sin editar, así como preguntas-respuestas del tipo *"Which command copies files?"*. Más recientemente, el sistema ha sido remodelado para los *Aircraft Maintenance Manuals (AMM)* del Airbus A320. La naturaleza altamente técnica de este dominio, un formato basado en SGML y un volumen de texto muy superior (120MB) al de la documentación de Unix fueron factores determinantes a la hora de demostrar la escalabilidad y la no dependencia de área de aplicación de nuestro sistema.

Si se le compara con los sistemas presentados en TREC, los dominios aquí vienen representados por colecciones de documentos de tamaños pequeño y mediano. No cabe duda que la oportunidad de poder procesar toda la colección de documentos en la fase fuera de línea, frente a sólo una selección de textos, supone una gran ventaja. Pero, por otro lado, a medida que las colecciones aumenten de tamaño este procedimiento pasará a ser computacionalmente demasiado costoso, por lo que será necesario añadir una indización de párrafos.

Las preguntas del usuario son procesadas en línea, convertidas en FLMs y comprobadas con la base de conocimiento de documentos. Además, la existencia de indicadores al texto original unidos a las formas lógicas extraídas contribuyen a que el sistema identifique y subraye esas palabras en la oración que conduce a la respuesta deseada (Mollá et al., 2000). Puede verse un ejemplo del *output* de ExtrAns en la figura 3.

Si no se halla una comprobación directa para la pregunta del usuario, el sistema es capaz de relajar los criterios de comprobación paso a paso. En primer lugar se añadirán hipónimos a los términos de la pregunta de modo que ésta sea más general, sin que por ello deje de ser correcta desde un punto de vista lógico. En caso de que esta medida falle, el sistema probará con un emparejamiento aproximado, basado en la extracción de la oración con mayor índice de coincidencia parcial de predicados, con la pregunta. Se puntúan entonces las oraciones emparejadas (parcialmente) y se proporcionan las que hayan alcanzado mejores resultados. Si este método no condujera a suficientes respuestas, el sistema probaría todavía con un emparejamiento mediante palabras clave en el que se dejan atrás los criterios sintácticos y sólo se utiliza la información acerca de categorías gramaticales (*"Part of Speech"*). Este último paso se asemeja a una metodología tradicional de recuperación de pasajes textuales basada en el etiquetaje de información sobre partes del discurso.

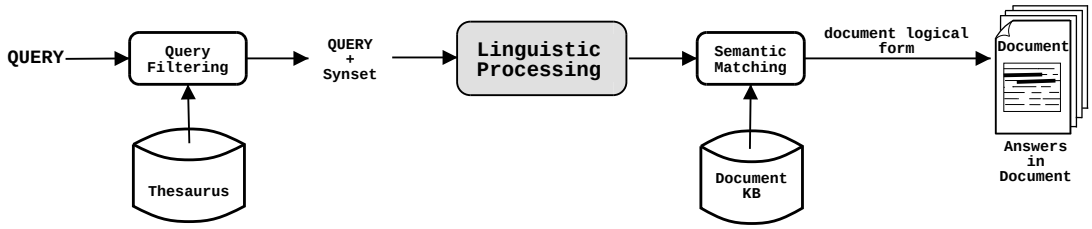

Figura 2: Arquitectura simplificada del sistema ExtrAns: procesamiento de las preguntas

3. *Procesamiento sintáctico*

El análisis sintáctico utiliza un analizador robusto llamado Link Grammar (LG) (Sleator y Temperley, 1993) basado en dependencias, que tiene capacidad para manejar gran variedad de estructuras sintácticas. Aquellas ambigüedades sintácticas irresolubles propias de la lengua inglesa, como las de complemento de sintagma preposicional o contrucciones de gerundio e infinitivo, se tratan con un procedimiento basado en corpus (Brill y Resnik, 1994). En cuanto a los pronombres intraoracionales (*"Sentence-internal pronouns"*), se recurre a un algoritmo de resolución de anáfora (Lappin y Leass, 1994).

La LG se basa en enlaces que describen la estructura sintáctica de una oración. Dichos enlaces conectan parejas de palabras de tal manera que los requisitos de cada palabra sean satisfechos, los enlaces no se crucen y las palabras conformen un grafo conectado.

A pesar de algunas extensiones a nivel léxico-semántico, el procesamiento de casos frecuentes de unidades multi-palabra de terminología específica supuso un problema para la LG. Pero con la añadidura de un nuevo módulo, capaz de identificar estos términos previamente detectados, los optimiza como unidades sintácticas propias, medida ésta que acaba reduciendo la complejidad del análisis en un 50 %. Por otro lado, se ha ampliado el *output* de la LG incluyendo la dirección de los enlaces ya que esta información es vital para las cuestiones de resolución de anáfora y el análisis semántico. Como la LG muestra todos los análisis posibles, hay que establecer un procedimiento para eliminar la ambigüedad entre ellos (Mollá y Hess, 2000), que permita identificar correctamente las relaciones de dependencia.

Estas relaciones de dependencia se emplean para generar la representación semántica de la oración. La LG posee un componente robusto que analiza estructuras complejas o agramaticales, de manera que ExtrAns pueda aún producir FLMs, ampliadas con predicados especiales que marcan las palabras no procesadas como "palabras clave".

El procesamiento de oraciones con nominalizaciones se apoya en la utilización de un pequeño léxico de nominalizaciones compilado manualmente, el cual nos ha permitido tratar los casos más importantes, ej. "to edit a text"/"editor of a text"/"text editor".

El sistema también incluye relaciones de hiponomia y sinonimia siguiendo el modelo de WordNet. Las relaciones de sinonimia son identificadas mediante la herramienta terminólogica Fastr (Jacquemin, 2001). Todas las palabras se asocian con su categoría gramatical, su raíz morfólogica y sus sinónimos. Cuando los términos conforman unidades multi-palabra, éstas se representan con una regla que aporta información sobre cada palabra en forma de característica-valor. De este modo, las metareglas presentan la relación de dos palabras, limitando su estructura como frase en función de la información morfológica y semántica que posee cada palabra. Todos los detalles se hallan en (Rinaldi et al., 2003).

Recientemente hemos conseguido que cada metaregla identifique la sinonimia estricta que resulta de las variaciones morfosintácticas (ej. "cargo compartment door"/"doors of the cargo compartment"), de los términos con núcleo sinónimo (ej. "electrical cable"/"electrical line"), así como aquellos con modificadores sinónimos (ej. "fastener strip"/"attachment strip") y ambos (ej. "functional test"/"operational check"). Para una descripción de la frecuencia y gran variedad de tipos de variación que se dan en el dominio del AMM, véase (Rinaldi et al., 2002a). Por otra parte, determinamos los casos de hiponimia léxica mediante un simple algoritmo de emparejamiento. Para más información, véase (Dowdall et al., 2002).

Cabe destacar que con el descubrimiento

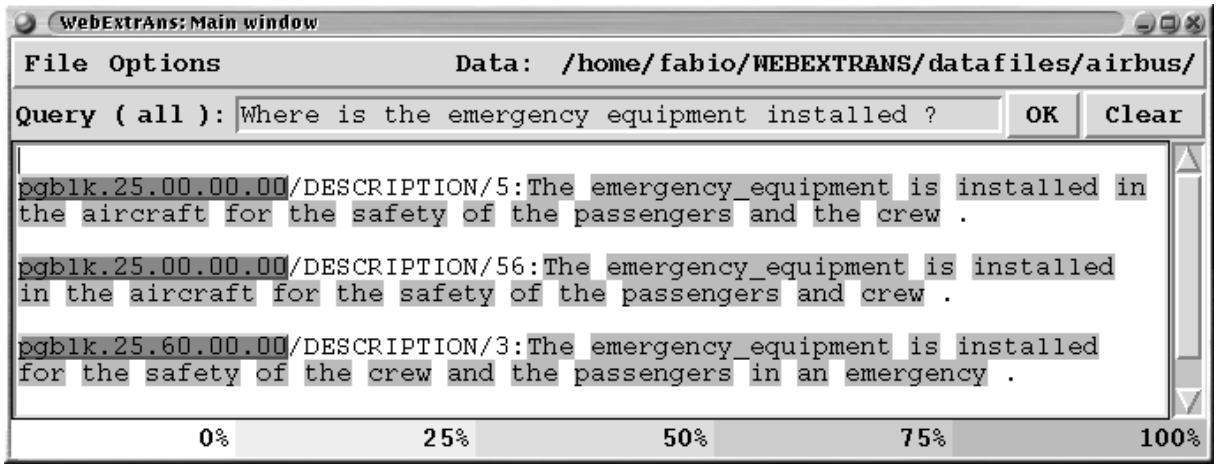

Figura 3: Ejemplo del *output* de ExtrAns, ventana de interrogación

automático de estas relaciones de sinonimia e hiponimia entre 6.032 términos del AMM ha producido 2770 *"synsets"* con 1176 enlaces de hiponimia. Posteriormente se ha llevado a efecto una inspección manual de 500 *"synsets"*, cuyo resultado presenta que sólo el 1,2 % de éstos tenía un término inapropiado. Por otro lado, una examinación similar de 500 enlaces hiponímicos ha concluído la validez de todos ellos.

4. *Análisis semántico*

Las Formas Lógicas Mínimas (FLMs) de los documentos y preguntas alcanzan su expresión semantica en ExtrAns. La generación de las FLMs es lo suficientemente robusta como para tratar oraciones muy complejas (incluso agramaticales) (Mollá et al., 2000) y facilita la comparación semántica de preguntas frente a documentos.

Un importante aspecto de las FLMs se deriva de las expresiones llanas producidas a través de la reificación, tal y como se propone por ejemplo en (Hobbs, 1985) o (Copestake, Flickinger, y Sag, 1997). De esta forma se pueden representar los modificadores de eventos, las negaciones, los verbos de primer orden, los condicionales y los predicados de orden superior .

Las FLMs usan las dependencias sintácticas principales entre palabras para expresar las relaciones verbo-argumento, así como las relaciones de modificador y adjunto. Por ejemplo, la FLM de la oración inglesa *"A coax cable connects the external antenna to the ANT connection?"*[2] es como sigue:

```
(1) holds(o1),
    object(coax_cable, o2,[v3]),
    object(external_antenna,o3,[v4]),
    object(ANT_connection,o4,[v5]),
    evt(connect,o1,[v3,v4]),
    prop(to,p1,[o1,v5]),
```

ExtrAns identifica tres términos multipalabra, representados en (1) como los complementos: v3, *coax_cable*; v4, *external_antenna*; v5, *ANT_connection*. La entidad $o1$ representa el evento *connect* que incorpora dos argumentos, *coax_cable* y *external_antenna*. Este evento reificado, $o1$, es utilizado de nuevo en la proposición final para describir que el evento $v5$ (*ANT_connection*) tiene lugar.

En esto consiste la utilidad del procedimiento de reificación: ofrecer los argumentos adicionales $o2$, $o3$, $o4$ y o1 como "enganche" para que éstos sean unidos a las entidades que denotan. La reificación puede utilizarse para incrementar monotónicamente la FLM que haya sido infraespecificada (1), sin incrustar argumentos (preservando una estructura llana), o reescribir destructivamente la FLM original.

Por ejemplo, la expresión inglesa *"A coax cable securely connects the external antenna to the ANT connection."*[3] no cambia para nada la FLM original, pero sí que por añadidura asevera con `prop(securely,p8,1)`, que el evento 1 es "secure".

[2] Un cable coax conecta la antena externa a la conexión ANT

[3] Un cable coax conecta de un modo seguro la antena externa a la conexión ANT

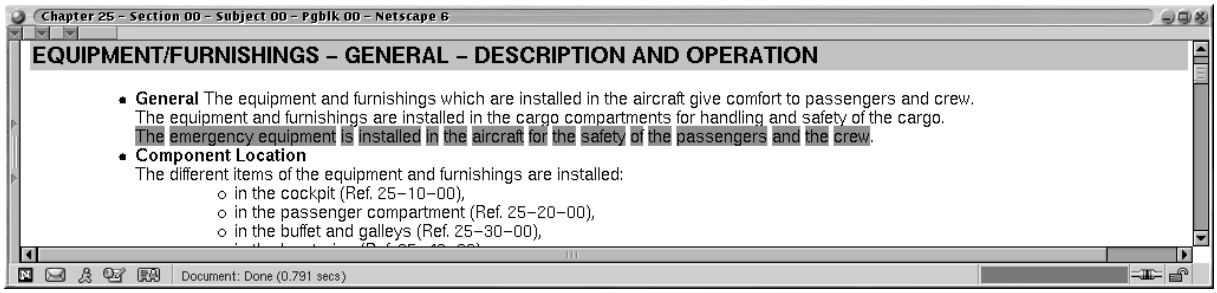

Figura 4: Ejemplo del *output* de ExtrAns, ventana del documento

5. Extracción de la Respuesta

ExtrAns encuentra las respuestas a las preguntas formando las FLMs de las preguntas y entonces hace correr el mecanismo de resolución con refutación de Prolog, a fin de hallar esas FLMs que mejor validen la pregunta. La forma lógica de la pregunta *"How is the external antenna connected?"*[4] sería:

```
(2) holds(v1),
    object(external_antenna,o2,[v5]),
    evt(connect,v1,[v4,v5]),
    object(anonymous_object,v3,[v4]).
```

Las variables introducidas en una FLM de pregunta se convierten en variables de Prolog. Así se puede correr la FLM resultante como una pregunta Prolog, que tendrá éxito si ha habido una aseveración en el texto que diga que la antena externa está conectada a o por algo (el objeto anónimo de la pregunta). Cada FLM está conectada a la oración de la cual deriva, lo que permite poder mostrarla despues al usuario como una respuesta (ver figura 3).

Mediante el uso de la resolución en Prolog se encontrarán las respuestas que de un modo lógico contesten la pregunta. No obstante dado que las FLMs no son más que formalismos lógicos simplificados convertidos en estructuras llanas, ExtrAns encontrará las oraciones que, desde un punto de vista lógico, puede que no sean respuestas exactas pero que sean aún relevantes para la pregunta del usuario.

6. Evaluación

A fin de llevar a cabo un ejercicio de evaluación para nuestro sistema, decidimos emplear un SRI como herramienta de cotejación, aún sabiendo que las medidas estándar de precisión y *recall* no son las ideales para un sistema de extracción de respuestas. Ciertamente, la *recall* tiene menor importancia que la precisión, puesto que el objetivo de tal sistema es dar (al menos) una respuesta correcta, más que proporcionar todas las respuestas aparentemente posibles a partir de una colección textual dada.

En la sección QA de TREC, una medida de precisión utilizada habitualmente es la denominada *"Mean Reciprocal Rank"* (MRR). El *rank* es la posición en la que se encuentra la primera respuesta correcta en la lista del *output* del sistema. Con un conjunto determinado de respuestas se computa el MRR como la media de las puntuaciones recíprocas otorgadas a todas las respuestas.

La evaluación que concretamente presentamos aquí va dirigida a la aplicación desarrollada para el dominio del AMM. Creamos 100 preguntas tras la cuidadosa selección de pasajes interesantes del manual y formulamos preguntas, cuya respuesta se podía hallar en dichos pasajes. Las preguntas fueron introduciéndose tanto a ExtrAns como al sistema de RI (SMART) que elegimos para el experimento. Mientras que en general ExtrAns genera un pequeño número de respuestas, las cuales pueden ser fácilmente comprobadas manualmente, SMART recupera una larga lista de documentos a los que les atribuye una posición (*ranking*). La inspección manual de todos los documentos recuperados por SMART habría sido imposible, por lo que se fijó un límite arbitrario de 10 documentos. Es decir, que si no se hallaba una respuesta válida en los primeros 10 documentos, clasificábamos la operación como "respuesta no encontrada".

El diagrama (5) muestra como se encuentran muchas respuestas por cada índice de

[4] ¿Cómo está conectada la antena externa?

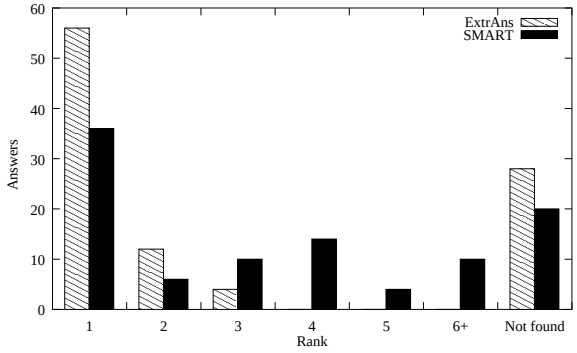

Figura 5: Respuestas a diferentes niveles (*ranks*)

puntuación (de 1 a 5, las respuestas de 6 a 10 se las considera juntas). Como puede apreciarse, ExtrAns encuentra menos respuestas que SMART (incluso descartando todas las respuestas con índice > 10). Por tanto, el grado de *recall* sería obviamente muy superior en SMART. Sin embargo, en la mayoría de los casos cuando ExtrAns encuentra la respuesta, le otorga la primera posición.

Para la evaluación que acabamos de relatar, nuestro sistema obtendría un MRR de 0.63, el cual es un resultado bastante bueno si lo comparamos con los resultados obtenidos en TREC. Sin embargo, habría que resaltar el hecho que una comparación puede conducir a conclusiones erroneas. En este sentido, nuestra evaluación es bastante más limitada que las que se acostumbran a realizar en TREC. Por otro lado, nuestro sistema no podría en estos momentos manejar unos volúmenes de texto comparables a los que se ven en TREC.

En términos generales, esta evaluación nos lleva a la conclusión que ExtrAns puede ofrecer una mayor precisión que cualquier sistema de RI, a expensas de una *recall* inferior. Sin embargo, sabemos que sólo ésta no nos interesaría puesto que en el escenario que nos ocupa lo importante es localizar rápidamente la respuesta precisa.

7. Discusión

Las técnicas de RI pueden ser utilizadas para implementar sistemas si son aplicadas a nivel de pasaje textual o de oración. Aquellas porciones de texto que tengan el máximo índice de coincidencia de términos aparecidos en la pregunta conducirán, con cierta probabilidad, a la respuesta. Los típicos pasos de preprocesamiento (eliminación de *"stop words"*, peso de palabras clave, etc.) se pueden utilizar para refinar este método básico. Sin embargo, los sistemas que no emplean técnicas de procesamiento lingüístico y que se limitan al procedimiento del *"bag of words"* heredado de la RI nunca podrán establecer una distinción entre los diferentes fragmentos textuales que contienen las mismas palabras en diferentes configuraciones sintácticas y, por tanto, portadoras de diferentes significados.

Los métodos estándar usados en la RI para categorizar aciertos en función de su relevancia tampoco son buenos substitutos para estas técnicas. La relevancia entendida en la RI se determina casi siempre en función a los pesos asignados a los términos individuales, los cuales se computan a partir de las frecuencias de los términos en los documentos (o pasajes) y en toda la colección de documentos (la variable expresada en inglés como *tf/idf*). Como esta medida no toma en consideración las relaciones sintácticas (y en consecuencia, semánticas), no puede distinguir entre aciertos que son lógicamente correctos y otros que no lo sean.

Resulta interesante observar cómo algunos de los que obtuvieron mejores resultados en TREC se han ido apartando progresivamente de los procedimientos tradicionales de RI e incorporando técnicas de procesamiento del lenguaje natural (PLN) que utilizan la información semántica. Por ejemplo el sistema con una mejor actuación en TREC 9 (Harabagiu et al., 2001) y TREC 11 (Moldovan et al., 2002) realiza un análisis completo de un conjunto de textos seleccionados para cada pregunta y de la misma pregunta, creando tras varios pasos intermedios una representacion lógica inspirada en la anotación de Hobbs.

8. Conclusión

Las competiciones TREC han puesto de manifiesto que no se puede prescindir de las técnicas de PLN si se pretende señalar las respuestas relevantes con precisión.

El significado de las preguntas y los documentos debe tomarse en consideración mediante un análisis sintáctico y semántico. Nuestro sistema de ER, ExtrAns, muestra que tales aplicaciones son una realidad viable con la tecnología actual.

Bibliografía

Abney, Steven, Michael Collins, y Amit Singhal. 2000. Answer extraction. En Sergei Nirenburg, editor, *Proc. 6th Applied Natural Language Processing Conference*, páginas 296–301, Seattle, WA. Morgan Kaufmann.

Brill, Eric y Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. En *Proc. COLING '94*, volumen 2, páginas 998–1004, Kyoto, Japan.

Copestake, Ann, Dan Flickinger, y Ivan A. Sag. 1997. Minimal recursion semantics: an introduction. Informe técnico, CSLI, Stanford University, Stanford, CA.

Dowdall, James, Michael Hess, Neeme Kahusk, Kaarel Kaljurand, Mare Koit, Fabio Rinaldi, y Kadri Vider. 2002. Technical terminology as a critical resource. En *International Conference on Language Resources and Evaluations (LREC-2002), Las Palmas*, 29–31 May.[5]

Harabagiu, Sanda, Dan Moldovan, Marius Paşca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Gîrju, Vasile Rus, y Paul Morarescu. 2001. FALCON: Boosting knowledge for answer engines. En Voorhees y Harman (Voorhees y Harman, 2001).

Hobbs, Jerry R. 1985. Ontological promiscuity. En *Proc. ACL'85*, páginas 61–69. University of Chicago, Association for Computational Linguistics.

Jacquemin, Christian. 2001. *Spotting and Discovering Terms through Natural Language Processing*. MIT Press.

Lappin, Shalom y Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Moldovan, Dan, Sanda Harabagiu, Roxana Girju, Paul Morarescu, Finley Lacatusu, Adrian Novischi, Adriana Badulescu, y Orest Bolohan. 2002. LCC Tools for Question Answering. En *Proceedings of TREC-11*.

Mollá, Diego y Michael Hess. 2000. Dealing with ambiguities in an answer extraction system. En *Workshop on Representation and Treatment of Syntactic Ambiguity in Natural Language Processing*, páginas 21–24, Paris. ATALA.

Mollá, Diego, Rolf Schwitter, Michael Hess, y Rachel Fournier. 2000. Extrans, an answer extraction system. *T.A.L. special issue on Information Retrieval oriented Natural Language Processing*.[5]

Rinaldi, Fabio, James Dowdall, Michael Hess, Kaarel Kaljurand, y Magnus Karlsson. 2003. The Role of Technical Terminology in Question Answering. En *Proceedings of TIA-2003, Terminologie et Intelligence Artificielle*, páginas 156–165, Strasbourg, April.[5]

Rinaldi, Fabio, James Dowdall, Michael Hess, Kaarel Kaljurand, Mare Koit, Kadri Vider, y Neeme Kahusk. 2002a. Terminology as Knowledge in Answer Extraction. En *Proceedings of the 6th International Conference on Terminology and Knowledge Engineering (TKE02)*, páginas 107–113, Nancy, 28–30 August.[5]

Rinaldi, Fabio, James Dowdall, Michael Hess, Diego Mollá, y Rolf Schwitter. 2002b. Towards Answer Extraction: an application to Technical Domains. En *ECAI2002, European Conference on Artificial Intelligence, Lyon*, 21–26 July.[5]

Rinaldi, Fabio, James Dowdall, Michael Hess, Diego Molla, Rolf Schwitter, y Kaarel Kaljurand. 2003. Knowledge-Based Question Answering. En *Knowledge-Based Intelligent Information and Engineering Systems (KES-2003)*, Oxford, September. Aceptado para publicación[5].

Sleator, Daniel D. y Davy Temperley. 1993. Parsing English with a link grammar. En *Proc. Third International Workshop on Parsing Technologies*, páginas 277–292.

Voorhees, Ellen M. 2000. The TREC-8 Question Answering Track Evaluation. En Ellen M. Voorhees y Donna Harman, editores, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*. NIST.

Voorhees, Ellen M. y Donna Harman, editores. 2001. *Proceedings of the Ninth Text REtrieval Conference (TREC-9), Gaithersburg, Maryland, November 13-16, 2000*.

[5] Disponible en: `http://www.cl.unizh.ch/CLpublications.html`