

# Algoritmos para la Generación de Consultas en SPARQL como Herramientas para el Uso de Tecnologías de la Web Semántica

C.M. Vázquez<sup>1,2</sup> y A. Montes<sup>2</sup>

<sup>1</sup>Centro de Desarrollo de Software, Universidad Tecnológica Emiliano Zapata del Estado de Morelos (UTEZ), Morelos, México

<sup>2</sup>Grupo de Ingeniería Web, Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), Morelos, México

## RESUMEN

*Considerando que la Web Semántica surge con la necesidad de proporcionar estructura a la información contenida en la Web actual, y que existe un brecha entre los usuarios de la Web y la manera de utilizar las tecnologías que forman las bases de la Web Semántica, en este artículo se presentan algoritmos que permiten traducir una consulta específica del usuario (lenguaje natural) en inglés al lenguaje de consulta para la Web Semántica: SPARQL, mediante la identificación y tratamiento de los fenómenos lingüísticos de la elipsis y genitivo sajón presentes en el lenguaje natural. El análisis de la consulta del usuario se realiza mediante la búsqueda de expresiones regulares para la detección de entidades (nombres) y acciones (verbos) con la finalidad de formar tripletas, las cuales serán utilizadas para construir la consulta en SPARQL.*

**Palabras clave:** Web Semántica, Fenómenos Lingüísticos, Procesamiento del lenguaje Natural

## ABSTRACT

*Whereas the Semantic Web comes with the need to provide structure to the information contained in the current web and there is a gap between Web users and how to use the technologies that form the basis of the Semantic Web, this paper presents algorithms to translate a user's query (natural language) to the query language for the Semantic Web: SPARQL, through the identification and treatment of linguistic phenomena of ellipsis and Saxon genitive present in natural language. The analysis of the user's query is done by regular expression search for the detection of entities (nouns) and actions (verbs) in order to form triplets, which will be used to construct the query in SPARQL.*

**Keywords:** Semantic Web, linguistic phenomena, Natural Language Processing

## 1. Introducción

La Web actual, a pesar de ser la mayor fuente de información a nivel mundial carece de una estructura semántica, lo que provoca que mucha de la información disponible sólo pueda ser interpretada en su mayoría por los humanos.

El proceso de agregar sentido (semántica) a los servicios y contenidos Web, ha sido aceptado como el siguiente paso de la evolución de la *World Wide Web*: la Web Semántica [BHL01], dejando de lado su concepción original como una Web de documentos.

La Web Semántica proporciona lenguajes que expresan la información contenida en ella en formato interpretable por las máquinas.

## 2. Motivación

Hoy en día han sido desarrolladas muchas tecnologías relacionadas a la Web Semántica. Uno de los avances más destacados son los lenguajes para el desarrollo de ontologías. La *World Wide Web Consortium* (W3C) ha aprobado estándares que brindan bases sólidas para la consoli-

dación de la Web Semántica, las cuales abrirán brechas para el desarrollo de aplicaciones de nueva generación [TASW05].

La Web Semántica se basa en el uso de ontologías en OWL [OWL12] y estructuras de tripletas RDF [RDF12] de la forma <sujeito, predicado, objeto> para lograr una representación abstracta de un dominio, y son éstas las que dan significado a la misma. Por tal motivo, se están desarrollando mecanismos de recuperación de información y sistemas de preguntas y respuestas que actúen sobre las bases de la Web Semántica (ontologías y rdf) mediante el lenguaje de consulta desarrollado para la misma denominado SPARQL [SPARQL12].

Desafortunadamente, hoy en día todavía existe una brecha entre las tecnologías de la Web Semántica y los usuarios, esto debido al desconocimiento del funcionamiento de los lenguajes formales que permitirán hacer uso de ellas.

Por lo anterior, en el presente trabajo se desarrollaron algoritmos con la finalidad de analizar las solicitudes de un usuario común (en lenguaje natural) y transformarlas a una representación en lenguaje SPARQL independiente del dominio de la base de conocimiento (ontologías y rdf).

### 3. Estado del arte

Se han realizado varios esfuerzos para disminuir la brecha existente entre las tecnologías de la Web Semántica y el usuario común. En el presente estado del arte se analizan los mecanismos utilizados por estos esfuerzos para que los usuarios comunes puedan hacer uso de las tecnologías de la Web Semántica sin necesidad de conocer los lenguajes formales para la misma.

En los trabajos de [LUSM10], [LUMP06] y [VCG09] se implementaron técnicas de procesamiento de lenguaje natural con la finalidad de interpretar la solicitud del usuario y recuperar información desde ontologías, sin considerar el lenguaje de recuperación recomendado por la W3C: SPARQL. Además, trataron el fenómeno lingüístico de la sinonimia mediante el enriquecimiento de la consulta añadiendo palabras semejantes a las búsquedas realizadas sobre las ontologías.

En los trabajos de [KBZ06], [FIFV08] y [RLL09] se desarrollaron algoritmos para transformar solicitudes de usuario a consultas en SPARQL, para cada solicitud del usuario se implementaron técnicas de procesamiento del lenguaje natural para dar tratamiento a los fenómenos lingüísticos de la sinonimia y elipsis.

Los algoritmos presentados en este trabajo consideran la representación intermedia basada en tripletas generadas a partir de patrones identificados mediante expresiones regulares, con la finalidad de traducir la consulta en lenguaje natural del usuario a una consulta en SPARQL, además de considerar el tratamiento de los casos genitivos y el fenómeno lingüístico de la elipsis, el fenómeno lingüístico de la sinonimia no se consideró dentro de los algoritmos debido a que al ser independientes del dominio, la única información con la que se cuenta es la solicitud del usuario y no recibe como entrada una ontología de dominio como los trabajos anteriores.

### 4. Algoritmos para el análisis de la consulta

Los siguientes algoritmos desarrollados trabajan con base a un patrón (secuencia de letras) generado a partir de un análisis léxico/sintáctico a la solicitud del usuario.

El análisis léxico/sintáctico etiqueta cada componente lingüístico (palabra) con su respectiva categoría gramatical. Considerando que el resultado de este análisis se basó en el estándar del *Penn Tree Bank Project* [PTP12], la Tabla 1 resume los símbolos que identificarán a cada categoría gramatical.

La mayoría de los analizadores disponibles en la red consideran el etiquetado de [PTP12] como base para la generación de sus resultados (análisis léxico/sintáctico), actualmente en [SGN12] el grupo para el Procesamiento del Lenguaje Natural de la Universidad de Stanford puso a disposición su analizador para la comunidad en general.

Si consideramos la siguiente consulta: *Which Azucena's students are working on ontologies and are generating ontologies of CENIDET?*, generará el patrón mostrado en la Figura 1.

Tabla 1. Símbolos asignados a cada una de las categorías gramaticales.

Categoría	Etiquetado	Símbolo
Palabras de preguntas	WP, WDT	Q
Nombres	NN, NNP, NNPS, NNS	N
Verbos	VB, VBD, VBG, VBN, VBP, VBZ	V
Adjetivos	JJ, JJR, JJS	A
Preposiciones	IN TO	P F(para posesivos) T
Determinante	DT	D
Conjunciones/Disyunciones	CC	C
Poseivos	POS	O

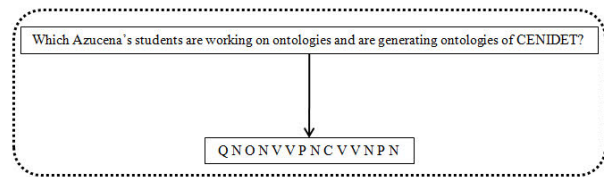


Figura 1. Patrón generado de la consulta del usuario

### 4.1 Algoritmo para el tratamiento del Genitivo Sajón

El genitivo sajón es conocido en ocasiones como el caso posesivo o la forma posesiva. El genitivo sajón se identifica mediante el uso de 's o 's exclusivamente para personas u organismos vivientes. La construcción 's no es posible en *the key of the door* o *the leg of the table* por la regla mencionada anteriormente [ALE99].

El algoritmo desarrollado para la detección e identificación de los casos genitivos se enuncia a continuación:

#### Algoritmo 1. Tratamiento del genitivo sajón

```

Sea P el patrón generado a partir de cada componente lingüístico en la consulta del
usuario, x el valor de cada elemento que integra a P, de tal forma que no existe una
x que no pertenezca a P ó lo que es lo mismo ¬(x ∈ P), e i la posición de cada x en
P, entonces:

Para cada x ∈ P hacer
  Si (dx | x[i] ⇒ "O") ⇒ [dx | x[i-1] ∧ dx | x[i+1] ⇒ "N"] entonces
    Mientras Bx | x[i-1] ⇒ "N"
      crear CadenaIzquierda ← Bx | x[i-1] → "N"
    Fin_Mientras

    Mientras Bx | x[i+1] ⇒ "N"
      crear CadenaDerecha ← Bx | x[i+1] → "N"
    Fin_Mientras

  Fin_Si
Fin_Para

Identificar y sustituir [ (CadenaIzquierda ∪ "O" ∪ CadenaDerecha) ∈ P ]
- (CadenaDerecha ∪ "P" ∪ CadenaIzquierda)
    
```

El funcionamiento del algoritmo anterior consiste en revisar cada elemento (letra) dentro del patrón de la consulta, en caso de encontrarse un símbolo "O" se identifica un

caso genitivo. El tratamiento de los casos genitivos consiste en recorrer el patrón de la consulta a la izquierda y a la derecha del símbolo “O” para obtener todos los símbolos “N” que representaran los sustantivos que realizan la acción de pertenencia.

Una vez extraídos los símbolos “N” se sustituye en el patrón de la consulta el símbolo “O” por el de “F” y los sustantivos extraídos intercambian su posición y se reformula la consulta.

La Figura 2 muestra el resultado de la identificación y tratamiento del caso genitivo en la consulta de ejemplo.

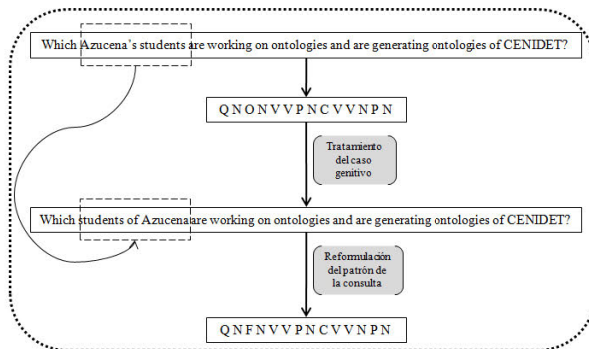


Figura 2. Tratamiento del caso genitivo y la reformulación de la consulta.

#### 4.2 Algoritmo para el tratamiento de la elipsis

La elipsis se hace presente comúnmente en el lenguaje natural debido al fenómeno de la economía de palabras, según [SÁN07] existen tres tipos: copulativas, disyuntivas y adversativas, de las cuales sólo se consideran las dos primeras.

El tratamiento de la elipsis implica la división del patrón (Figura 3) generado a partir de la consulta del usuario por cada conjunción coordinante (símbolo C) generando subpatrones, es importante que el genitivo sajón (en caso de haberlo) haya sido resuelto antes de tratar este fenómeno.

Se identifican dos casos del fenómeno lingüístico de la elipsis: con sujeto implícito y con sujeto explícito.

El primer caso permite reconstruir consultas que contengan el sujeto (¿de quién se pregunta?) implícito en la palabra de consulta, como serían solicitudes con la palabra de pregunta *Who*. El Algoritmo 2 permite identificar y tratar este tipo de caso se enuncia a continuación.

El segundo caso permite reconstruir consultas que indiquen de forma explícita la información solicitada. El algoritmo 3 permite identificar y tratar este tipo de caso se enuncia a continuación.

Los algoritmos para el tratamiento de la elipsis trabajan a partir de los subpatrones en los cuales se ha dividido el patrón de la consulta del usuario, a partir del segundo subpatrón se realiza una comparación con el primero y determina qué información del primer subpatrón debe ser agregada al subpatrón en comparación para formar una consulta completa.

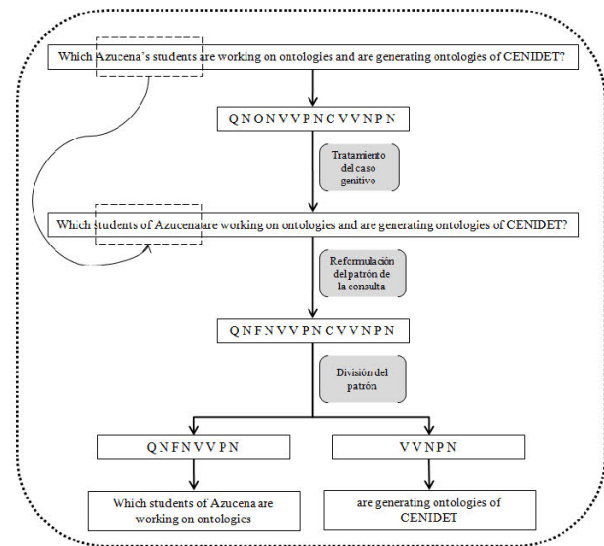


Figura 3. División del patrón de la consulta del usuario

#### Algoritmo 2. Tratamiento de la elipsis (caso “Who”)

Sea  $i$  el índice de cada uno de los subpatrones  $Q[i]$  generados de la consulta del usuario comenzando por 1, de tal forma que  $Q[1] \cup Q[1+1] \cup Q[1+2] \cup \dots \cup Q[n]$  corresponden a la consulta original ( $Q$ ) del usuario. Sean las expresiones regulares definidas como:

```
R1  (^((V)+(A)(0,1)(P|T)(0,1)))
    Identifica a todo patrón que inicie con un verbo, seguido de un o
    ningún adjetivo (o adverbio), seguido de una o ninguna preposición
    (éstas pueden venir etiquetadas con los símbolos P y T).

R2  (^([Q|V](0,1)A*N*F*N*))
    Identifica a todo patrón que inicie con una palabra de consulta
    (what_?, which_?, etc.) o un verbo (Is_?, Do_?, etc.), seguido de un
    adjetivo, nombre(s), preposición y nombre(s) con repeticiones de cero a
    muchas veces.

R3  (^Q)
    identifica a todo patrón de inicie con una palabra de consulta.
```

Que representan un conjunto infinito de patrones con índice  $k$ . Entonces:

```
Desde  $i+1$  hasta  $n$  hacer
  Si  $Sk \mid R1[k] \subset Q[i]$  entonces
    Si  $(Q' = Q[i]) \wedge [parteDe("Who", Q)]$  entonces
       $Q = Q[1] \cup 'C' \cup \{Sk \mid R3[x] \subset Q[1]\} \cup Q[i]$ 
    Sino
       $Q = Q[1] \cup 'C' \cup \{Sk \mid R2[x] \subset Q[1]\} \cup Q[i]$ 
  Fin_Si
Fin_Desde
```

El tratamiento de la elipsis utilizando el algoritmo anterior genera tantas consultas simples como conjunciones presentes en la consulta del usuario, se basa en la idea de ser menos complicado analizar oraciones simples a diseñar mecanismos que trabajen con estructuras complejas.

La Figura 4 muestra el resultado de la aplicación del algoritmo a la consulta del usuario.

#### 5. Proceso para la generación de tripletas

La generación de tripletas de la consulta tiene lugar una vez que la consulta del usuario no presenta los fenómenos lingüísticos del genitivo sajón o elipsis (no presentaba o

**Algoritmo 3.** Tratamiento de la elipsis (caso "Wh")

```

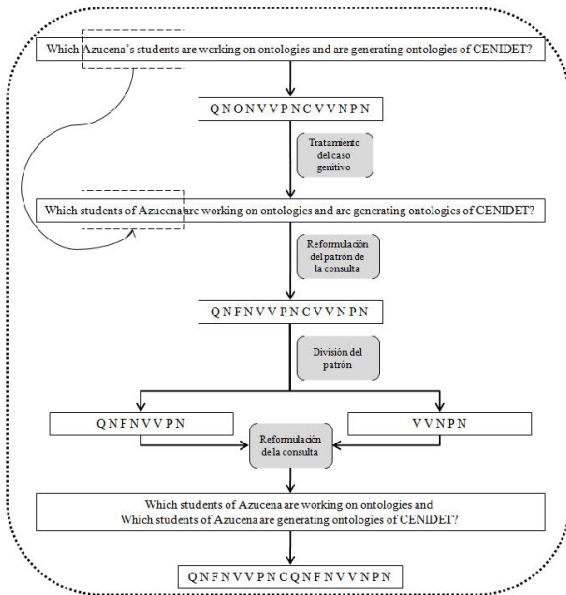
Sea i el índice de cada uno de los subpatrones Q[i] generados de la consulta del usuario
comenzando por 1, de tal forma que Q[1] ∪ Q[1+1] ∪ Q[1+2] ∪ ... Q[n] corresponden a la
consulta original (Q) del usuario. Sean las expresiones regulares definidas como:

R1  (^([P]{0,1}[D]{0,1}[A]*[N]+))
Identifica a todo patrón que inicie con una o ninguna preposición y/o
determinante, seguidos de un o ningún adjetivo, seguidos de por lo
menos un nombre.

R2  (^([Q|V]{0,1}[A]*[N]*[P]*[V]*))
Identifica a todo patrón que inicie con una palabra de consulta (what_?,
which_?, etc.) o un verbo (is_?, do_?, etc.), seguido de un
adjetivo, nombre(s), preposición y nombre(s) con repeticiones de cero a
muchas veces.

R3  (^([Q|V]{0,1}[A]*[N]*[P]*[V]*))
Identifica a todo patrón que inicie con una palabra de consulta (what_?,
which_?, etc.) o un verbo (is_?, do_?, etc.), seguido de un
adjetivo, nombre(s), preposición (generada a partir del tratamiento del
genitivo sajón), nombre(s) y preposición con repeticiones de cero a
muchas veces.

Que representan un conjunto infinito de patrones con índice k. Entonces:
Desde i+1 hasta n hacer
  Si Ek | R1[k] ⊆ Q[i] entonces
    Si 'P' ∈ Q[i] entonces
      Q = Q[1] ∪ 'C' ∪ {Ek | R2[k] ⊆ Q[1]} ∪ Q[i]
    Sino
      Q = Q[1] ∪ 'C' ∪ {Ek | R3[k] ⊆ Q[1]} ∪ Q[i]
  Fin_Si
Fin_Desde
    
```



**Figura 4.** Tratamiento de la elipsis y reformulación de la consulta del usuario.

fueron resueltos), por lo tanto, la consulta actual podría presentarse de dos maneras: como una consulta simple o como una consulta compuesta que consta de dos o más consultas simples unidas por conjunciones.

Las expresiones regulares de la Tabla 2 se combinan para formar expresiones más complejas con la intención de identificar los componentes de las tripletas <sujeeto, predicado, objeto>.

Una de las tripletas que se detectan a partir de las expresiones regulares, resulta de la relación desconocida entre dos nombres unidos mediante una preposición. Considere-

**Tabla 2.** Categorías de expresiones regulares

Categoría	Expresión regular	Descripción
<b>Nombres</b>	[D]{0,1}[A]*[N]+	Identifica la formación de nombres que pueden comenzar con cero o un determinante seguido de cero o muchos adjetivos y de al menos un nombre.
<b>Preposiciones</b>	[P F]+	Identifica las preposiciones, las que son etiquetadas como P son aquellas identificadas desde el primer análisis de la consulta, mientras que las etiquetadas como F se generan cuando se resuelve un caso genitivo.
<b>Verbos</b>	(V)+(A P T){0,1}	Identifica la existencia de al menos un verbo, el cual puede ir precedido por un adjetivo adverbial, o preposición con cero o una ocurrencia.

rando la frase "ontologies of CENIDET" extraída de la consulta de ejemplo se genera la siguiente tripleta: <ontologies, Unknown, CENIDET>, en la cual la palabra Unknown expresa la relación desconocida entre ambas palabras.

Las tripletas generadas de la consulta de ejemplo se muestran en la siguiente Figura 5.

Las tripletas [students,Unknown,Azucena] y [students ?,is-a,students] se encuentran repetidas debido a que se generan de cada una de las consultas simples (en el ejemplo presentado son dos consultas simples). El símbolo "?" en un uno de los componentes de la tripleta indica el elemento buscado.

**6. Proceso para la generación de la consulta SPARQL**

El proceso para la generación de la consulta en SPARQL tiene por objetivo representar las tripletas formadas a partir de la consulta en lenguaje natural del usuario, en una secuencia de sentencias que conformarán la consulta en SPARQL.

Se consideró la construcción de dos tipos de consultas en SPARQL dependiendo de la forma de la solicitud del usuario, estas son: SELECT y ASK. La consulta en SPARQL tendrá definidos tres prefijos, cuyos valores (URI's) se generan de forma automática cuando se crean ontologías

```
[ students ? , is-a , students ]
[ students ? , is-a , students ]
[ students , Unknown , Azucena ]
[ students , Unknown , Azucena ]
[ ontologies , Unknown , CENIDET ]
[ students , are working on , ontologies ]
[ students , are generazing , ontologies ]
```

**Figura 5.** Tripletas generadas a partir de la consulta del usuario.

desde un editor, estos prefijos son: rdf, rdfs y owl.

La construcción de la consulta en SPARQL se basa en el uso de plantillas. Las consultas SELECT devuelven como resultado el valor solicitado por el usuario y se mostrarán tantos resultados como existan tripletas en la ontología que coincidan con las tripletas generadas a partir de la consulta. Las consultas ASK devuelven como resultado afirmaciones o negaciones después de comparar de manera exacta las tripletas de la ontología contra las generadas a partir de la consulta del usuario.

Con base en las tripletas generadas de la consulta del usuario se determina que la consulta SPARQL necesaria es de tipo SELECT, por lo que las tripletas son agregadas a la plantilla correspondiente y el resultado de la consulta en SPARQL se muestra en la siguiente figura:

```
*PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
*PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
*PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT DISTINCT students ?
WHERE {
  students ? , is-a , students
  students , Unknown , Azucena
  students , Unknown , Azucena
  ontologies , Unknown , CENIDET
  students , are working on , ontologies
  students , are generazing , ontologies
}
```

**Figura 6.** Consulta SPARQL generada a partir del análisis de la solicitud del usuario.

## 7. Pruebas

El proceso de generar consultas en SPARQL a partir de consultas del usuario en lenguaje sin necesidad de conocer la fuente de información, que en este caso sería las ontologías, contribuye al desarrollo y aplicación de tecnologías para la Web Semántica.

Los algoritmos y procedimientos presentados fueron probados de manera inicial sobre un banco de preguntas desarrollado en [KMI12] como parte de un repositorio de prueba para un sistema desarrollado por ellos. Este repositorio cuenta con 16 categorías de consultas basadas en criterios lingüísticos puros. Las categorías de preguntas probadas con los algoritmos fueron:

- Categoría 1: Wh-genericterm
- Categoría 2: Wh-unknrel

- Categoría 5: Affirmative-negative
- Categoría 6: Affirmative-negative-pseudorel
- Categoría 7: Wh-3term
- Categoría 8: Wh-generic-1termclause
- Categoría 12: Wh-comb-and
- Categoría 13: Wh-com-or

Un total de 105 preguntas se extrajeron del banco de prueba, y los resultados de la aplicación de los algoritmos se resumen en la siguiente tabla:

**Tabla 3.** Resultado de las pruebas sobre el banco de preguntas.

Preguntas totales	105
Preguntas eliminadas	29
Preguntas consideradas	76
Éxitos	65
Fracasos	11
Eficiencia	85.52%

Los resultados presentados muestran un 85.52% de efectividad, los casos fallidos se debieron principalmente a errores durante el análisis léxico/sintáctico arrojados por la herramienta Stanford Parser, de cualquier forma sólo representan el 14.47% del total de las preguntas. Las preguntas eliminadas fueron aquellas que no pueden ser analizadas por algoritmos, debido a que éstos fueron desarrollados considerando el grado de expresividad del lenguaje SPARQL, el cual por poner un ejemplo, no cuenta con una función que me permita contar el número de registros devueltos a partir de una consulta con la palabra "how".

## 8. Conclusiones

Mediante el uso de herramientas de análisis léxico/sintáctico, y a partir de un tratamiento del lenguaje natural basado en expresiones regulares para la identificación de patrones, es posible representar la consulta del usuario en formato de tripletas <sujeito, predicado, objeto> y éstas se organizan para formar una representación de consulta en SPARQL.

La identificación y tratamiento de casos genitivos y el fenómeno de la elipsis se realizó mediante la generación de un conjunto de reglas (algoritmos), estas reglas consideran parte de las variantes gramaticales en las que se presentan estos fenómenos con la finalidad de generar consultas sin información gramatical faltante a partir de la solicitud del usuario expresada lenguaje natural.

El proceso de análisis de la consulta del usuario se basó en la generación de tripletas como una representación intermedia de la misma, muchos trabajos habían concebido la idea de construir tripletas pero en base a la ontología a consultar, la metodología desarrollada en el presente trabajo no considera una ontología base, por lo que fue neces-

rio modificar e integrar algunas de las técnicas revisadas, y generar algoritmos para construir tripletas sólo con la información proporcionada por el usuario en su consulta.

La generación de la consulta en SPARQL se realiza a partir de las tripletas generadas, la cual puede tener una estructura con sentencia SELECT o ASK considerando sólo la estructura e información de la consulta del usuario en lenguaje natural.

El resultado de este trabajo muestra las siguientes ventajas en comparación con otras herramientas semejantes presentadas en el estudio del estado del arte:

- El uso de expresiones regulares para la identificación de patrones, permite detectar y extraer las entidades (nombres) y acciones (verbos) en la consulta en lenguaje natural sin la necesidad de acceder y hacer inferencias con las entidades de la ontología.
- El tratamiento de fenómenos lingüísticos o peculiaridades de la gramática inglesa, con la finalidad de reconstruir la consulta en lenguaje natural para analizarla sin necesidad de la intervención del usuario.
- La construcción de consultas en SPARQL con sentencias SELECT y ASK, a partir de las tripletas generadas mediante la identificación de patrones con expresiones regulares.

Actualmente los algoritmos se están probando sobre una aplicación Web, y considerando no sólo técnicas de lenguaje natural sino también técnicas de alineamiento de ontologías como mecanismo de tratamiento del fenómeno lingüístico de la sinonimia.

## Referencias

- [ALE99] ALEXANDER L. *Longman English Grammar*. Editorial Longman, 1999.
- [BHL01] BERNERS-LEE T., HENDLER J., LASSILA O.: The Semantic Web. *Scientific American* (2001), vol. 284, pp. 33-43.
- [FIFV08] FERRÁNDEZ O., IZQUIERDO R., FERRANDEZ S., VICEDO J.: QACID: Un sistema de búsqueda de respuestas basado en ontologías, implicación textual y entornos reales. *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural* (2008), vol. 41, pp. 47-54.
- [KBZ06] KAUFMANN E., BERNSTEIN A., ZUMSTEIN R.: Querix: A natural language interface to query ontologies based on clarification dialog. *5ta International Semantic Web Conference* (2006). Atenas, Grecia.
- [KMI12] Knowledge Media Institute. Aqualog (2012), [online], <http://technologies.kmi.open.ac.uk/aqualog/>.
- [LUMP06] LÓPEZ V., UREM V., MOTTA E., PASIN M.: AquaLog: An ontology-driven Question Answering system as an interface to the Semantic Web. *Human Language Technology Conference of North America Chapter of the Association of Computational Linguistics Proceedings* (2006), Nueva York, USA.
- [LUSM10] LÓPEZ V., UREM V., SABOU M., MOTTA E.: Cross ontology query answering on the semantic Web: An initial evaluation. *5ta International Conference on Knowledge Capture* (2010). California, USA.
- [OWL12] WORLD WIDE WEB CONSORTIUM. *Web Ontology Language* (2012). Disponible en: <http://www.w3.org/owl/>.
- [PTP12] PENN TREEBANK PROJECT. *Penn Treebank II Tags* (2012). Disponible en: <http://bulba.sdsu.edu/jeanette/thesis/PennTags.html>
- [RDF12] WORLD WIDE WEB CONSORTIUM. *Resource Description Framework* (2012). Disponible en: <http://www.w3.org/RDF/>.
- [RLL09] RAMACHANDRAN V., LLANGO K.: NLION: Natural Language Interface for Querying Ontologies. *Proceedings of the 2do Bangalore Annual Computer Conference* (2009), Bangalore, India.
- [SÁN07] Sánchez B. F.: *Gramática Inglesa*. Editorial Pearson Longman, 2007.
- [SNG12] STANFORD NLP GROUP. *Stanford Parser Demo* (2012). Disponible en: <http://nlp.stanford.edu:8080/parser/>.
- [SP12] STANFORD PARSER GROUP. *The Stanford Natural Language Processing Group* (2012). Disponible en: <http://nlp.stanford.edu/software/lex-parser.shtml>.
- [SPARQL12] WORLD WIDE WEB CONSORTIUM. *SPARQL Query Language for RDF* (2012). Disponible en: <http://www.w3.org/2001/sw/wiki/SPARQL>.
- [TASW05] TJOA A., ANDJOMSHOAA A., SHAYEGANFAR F. WAGNER R.: Semantic Web Challenges and New Requirements. *16th International Workshop on Database and Expert Systems Applications DEXA'05* (2005), pp. 86-94.
- [VCG09] VIVANCOS P., CASTEJÓN J., GARCÍA D.: INVENIO: Búsqueda semántica basada en tecnología Google. *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural* (2009), vol. 36, pp. 353-354.