Consultas Flexibles sobre bases de conocimiento extraídas desde la Web

Eduardo San Martin Villarroel[1] y Clemente Rubio Manzano[2]

Resumen

La web concentra una enorme cantidad de datos en su mayoría semi-estructurados. En este *paper* planteamos que es posible hacer uso de esos datos si contamos con los mecanismos apropiados de extracción, procesamiento y consulta. Para tal efecto, proponemos una integración de tres tecnologías (Programación lógica-difusa para realizar consultas flexibles, XML para almacenamiento y transporte de datos, y *jarvesting* como mecanismo de extracción) que derivan en una implementación práctica para efectuar consultas flexibles sobre bases de conocimiento construidas con información extraída desde la web.

Palabras claves: consultas flexibles, la minería web, programación lógica difusa, XML.

Flexible querying on knowledge bases extracted from the Web

Abstract

The Web contains an enormous number of mostly-semi-structured data. In this article, we suggest that it is possible to make use of these data if we have the appropriate mechanism of extraction, processing and querying. In order to achieve this, we propose an integration of three technologies (logic-fuzzy programming to make flexible queries, XML for data storing and transporting, and harvesting as an extraction mechanism) which results in a practical implementation to make flexible queries on knowledge bases built using data extracted from the Web.

Keywords: flexible queries, web mining, fuzzy logic programming, XML.

1 Dep. I+D, Fundación FINDESS, Camino las Mariposas km. 11, Chillan, Chile, zip code 3780000, teléfono (+569)74971514, email: eduardo.sanartin@findess.org
2 Dep. de Sistemas de Información, Univ. Del Bío-Bío, Avenida Collao Nº 1202, Concepción, Chile, zip code 4050231, teléfono (+5641) 2731258, email: clrubio@ubiobio.cl

Introducción

En la web podemos encontrar una enorme cantidad de datos de diversa índole. Muchas empresas u otros organismos hacen uso de ellos para obtener conocimiento de algún área y tomar decisiones. Muchos de esos datos se extraen desde la web usando diversas técnicas y mecanismos para luego ser procesados y utilizados en procesos ETL (*Extract, Transform and Load*), aplicaciones de inteligencia de negocios o consulta directa entre otros.

Para que los datos extraídos puedan ser transportados a distintos software o plataformas, generalmente se usan archivos XML (del inglés eXtensible Markup Language) (W3C), pues facilita la integración, el intercambio y la consulta de estos datos. Para consultar los datos, una alternativa es que los archivos XML que los contienen constituyan los orígenes de datos de sistemas de procesamiento y/o almacenamiento como Sistemas Administradores de Bases de Datos (SABD o DBMS en inglés) o aplicaciones de Inteligencia de Negocios (BI en inglés). Otra alternativa es efectuar consultas directamente sobre los datos extraídos, es decir, sobre los archivos XML sin necesidad de ingresarlos a otro sistema. Esta segunda opción es la que abordaremos en este paper.

Existen algunos mecanismos como extensiones de SQL (por sus siglas en inglés Structured Query Language) (Date & Darwen, 1997) que permiten efectuar consultas con documentos XML (Pankowski, 202) (Eisenberg & Melton) (YOSHIKAWA & AMAGASA, 2001). Estos lenguajes de consulta se han diseñado para recuperar información a partir de bases de datos que contienen datos precisos. Para que el usuario pueda efectuar consultas sobre los datos, debe proporcionar un conjunto de condiciones de selección que requieren comandos estrictos con parámetros y valores precisos. En muchas situaciones, los usuarios quienes requieren consultar los datos no están completamente familiarizados con estos, por lo tanto, es habitual que una consulta sea formulada por un experto que traduce una pregunta en lenguaje natural en restricciones precisas sobre valores rígidos, lo que implica la posibilidad de omitir respuestas de interés. Por otro lado, una parte de los conocimientos no se incorpora de una manera apropiada en los sistemas de almacenamiento de hoy en día (Tahami, 1977), ya que la información del mundo real a menudo es permeada por la vaguedad y/o imprecisión. La rigidez de los mecanismos de consulta dificulta la recuperación de información de manera flexible. En este sentido, las técnicas basadas en la teoría de conjuntos difusos (Zadeh., 1965) son muy útiles para modelar la vaguedad e imprecisión. Los conjuntos difusos pueden ser utilizados en el proceso de consulta de bases de datos para recuperar los datos de estos sistemas mediante consultas que contienen términos lingüísticos (Hsiao, 2007). En este trabajo se propone un método que permita realizar consultas flexibles sobre información extraída desde la web usando el estándar XML como mecanismo de transporte de datos y el modelamiento de la imprecisión mediante la construcción de una Base de Datos Borrosa con soporte para dos tipos de consultas flexibles: consultas lingüísticas difusas y consultas flexibles basadas en proximidad. Esta propuesta

nos permitirá recuperar datos de archivos XML usando el mecanismo de inferencia de un lenguaje de programación lógica basado en proximidad llamado Bousi~Prolog (P. Julián, 2009), a través del cual se realizarán las consultas flexibles sobre información extraída desde la web. Para ello, propondremos los mecanismos de extracción de datos desde la web para obtener un conjunto de registros almacenados en un documento XML. Luego se construirá una pequeña herramienta que permita la trasformación de documentos XML a predicados Bousi~Prolog con la posibilidad de establecer las variables lingüísticas que serán utilizadas en las consultas. Finalmente la imprecisión inherente de las variables lingüísticas será modelada mediante conjuntos borrosos y ecuaciones de proximidad en Bousi~Prolog. El resultado es un sistema que permita realizar consultas flexibles sobre información extraída desde la web.

Aplicación de la Lógica Difusa y Trabajo Relacionado

Intuitivamente, la imprecisión y la vaguedad aparecen cuando no sabemos qué valor exacto elegir para un atributo. Un conjunto difuso (Zadeh., 1965) se puede utilizar para representar información vaga y/o imprecisa. Existen dos líneas de trabajo en la utilización de conjuntos difusos aplicados a las bases de datos, esto es de particular interés, ya que nuestra fuente de datos será un documento XML que puede ser visto como una base de datos. En la primera línea de trabajo el modelo de datos se deja intacto y el procesamiento de consulta se ha extendido para permitir conceptos vagos que están representados por conjuntos difusos y/o variables lingüísticas (Kannan, 2010) (Takahashi, 1995) (Tahami, 1977). En la segunda línea de trabajo se aplica la teoría de conjuntos difusos para modificar el modelo de datos relacional (Petry, 1985) (Orchard, 1998) (Melton, 1999). El primer planteamiento parece ser más práctico y prometedor ya que el modelo relacional sigue siendo la alternativa más utilizada. Otros enfoques adoptan técnicas de agrupamiento como una herramienta para generar un mapeo entre los términos difusos, definidos como un nivel de abstracción mayor, y el registro de la base de datos (Hsiao, 2007)(al, 1990). Mientras que la primera opción requiere de una especificación exacta del significado de cada término difuso conocido por el sistema, el segundo hace que la tarea de definición sea más fácil ya que en lugar de definir explícitamente el rango de valores conceptuales correspondientes a cada término, el usuario solo define el orden relativo de los términos lingüísticos, y el sistema, mediante el algoritmo de clustering hará coincidir cada término lingüístico con el adecuado grupo de registros (al, 1990).

El trabajo presentado en este documento es un enfoque híbrido donde se emplean ambos mecanismos. En este contexto, el desarrollo de un sistema de consulta difusa sobre información extraída desde la web consistirá en la construcción de elementos de dos tipos. Primero, mecanismos que permitan la extracción y representación de información bajo el estándar XML y por otra parte, una interfaz de usuario que permita la transformación de los registros del documento XML en predicados Bousi~Prolog junto con la especificación

de las variables lingüísticas que serán utilizadas en: la construcción de relaciones entre los conceptos lingüísticos y los datos precisos del documento XML y la definición de relaciones de proximidad.

También es importante determinar el tipo de términos difusos. Consideramos que un término difuso se puede clasificar como: un descriptor numérico cualitativo, un descriptor no numérico cualitativo o una descripción de cuantificación. Un descriptor numérico cualitativo es una palabra que describe algún valor numérico o un rango de valores numéricos. Un descriptor cualitativo no numérico es una palabra que describe un concepto no numérico. Una descripción de cuantificación es una palabra que describe la cantidad de respuestas deseadas en una consulta en lenguaje natural que se refiere únicamente a los descriptores numéricos cualitativos. Nuestro enfoque puede funcionar tanto con descriptores no numéricos y numéricos cualitativos, ya que se basa en la programación lógica basada en proximidad, que es capaz de hacer frente a los dos tipos de descriptores.

Con el fin de tratar un descriptor numérico cualitativo, vamos a utilizar las técnicas desarrolladas para la construcción de un algoritmo de unificación semántica, que pondrá en relación un descriptor numérico con (posiblemente) varios conceptos lingüísticos, es decir, nuestro método actúa como un algoritmo de agrupamiento difuso.

Definiciones Preliminares, Conceptos y notaciones

Nuestra propuesta contempla una combinación de conceptos como: lógica difusa (Zadeh., 1965), programación lógica (Lloyd, 1987) y estándar XML (W3C). No obstante existe una cercanía importante entre XML y las base de datos relacional (Hsiao, 2007), por lo que junto a los conceptos mencionados, se expone también brevemente el modelo relacional.

XML y el modelo de datos relacional

En un modelo de datos relacional, una tabla es la estructura principal que representa una entidad del mundo real. Una tabla se define como un conjunto de columnas que corresponden a atributos de la entidad modelada. Cada entidad es representada por una fila (tupla) en la tabla. Formalmente una tabla T es un conjunto de tuplas y cada tupla es un conjunto de pares áatributo, valorñ de la forma t = {áA1, v1ñ, áA2, v2ñ,..., áAn, vnñ tal que vi Î Di y Di es un dominio. En este contexto, XML como metalenguaje puede ser utilizado para representar entre otras cosas, la estructura de una tabla de datos y contener al mismo tiempo los datos mediante elementos (estructura básica consistente de una etiqueta inicial, atributos y contenido opcionales, y una etiqueta final) (W3C). La estructura de un elemento es como sigue: `<nomEle [A1="v1",..., An="vn"]>[Contenid]</nomEle>`, donde `nomEle` es el nombre del elemento, `A1` es un atributo opcional, `vi` es el valor del atributo y Contenido es un contenido opcional del elemento. Un documento XML debe tener un único elemento raíz y los elementos dentro de otro elemento son elementos hijos. Un ejemplo de un documento XML bien formado es el

que se muestra en la figura 1.

En este documento podemos observar la coherencia con el modelo relacional donde la tabla T (<lista-vehículos> </lista-vehículos>) está formada por un conjunto de columnas (elementos hijos de <vehículo></vehículo>) las que a su vez están formadas por un conjunto de pares áatributo, valorñ donde atributo es un elemento y valor es el contenido del elemento. Cada elemento <vehículo></vehículo> corresponde una tupla de la tabla.

Relaciones Difusas y Dominios Sintácticos

El concepto de relación difusa fue introducido por Zadeh en (Zadeh, 1965). Una relación difusa binaria en un conjunto U es un subconjunto difuso en U x U (es decir, un mapeo de U x U ® [0, 1]). Se dice que una relación difusa binaria R es una relación de proximidad si satisface la propiedad reflexiva (es decir, R (x, x) = 1 para cualquier x Î U) y la propiedad simétrica (es decir, R (x, y) = R (y, x) para cualquier x, y Î U).

Si además se tiene la relación transitiva (es decir, R (x, z) ³ R (x, y) D R (y, z) para cualquier x, y, z Î U; donde el operador 'D' es una t-norma arbitraria), se le llama relación de similitud. Estamos principalmente interesados en las relaciones de proximidad sobre un dominio sintáctico.

FIGURA 1. EJEMPLO DE DOCUMENTO XML

```
<lista-vehículos>
 <vehículo>
  <marca>Toyota</marca>
  <anio>2013</anio>
  <precio>$ 12.600.000</precio>
  <motor>
   <cc>1600</cc>
   <tipo>En línea</tipo>
  </motor>
 <color>Rojo</color >
 </vehículo>
</lista-vehículos>
```

```
<Tabla>
 <tupla-1>
  <atributo-1>[valor]</ atributo-1>
   [<atributo-1.1>
    <atributo-1.n>[valor]</atributo-1.n>
   </atributo-1.1>]
 <atributo-n>[valor]</atributo-n>
 </tupla-1>
</Tabla>
```

Programación lógica difusa y Bousi~Prolog

La programación lógica difusa (Lee, 1972) introduce conceptos de lógica difusa en la programación lógica con el fin de hacer frente al tratamiento de la vaguedad de una manera declarativa. Cuando la imprecisión se modela mediante relaciones de similitud (Zadeh., 1965) entonces tenemos que hablar de programación lógica basada en similitud (Sessa, 2002). Bousi~Prolog (BPL, por sus siglas) (Rubio, 2009)(P. Julián, 2009) es una extensión del lenguaje Prolog estándar consistente en un marco de programación lógica basada en proximidad. Su semántica operacional es una adaptación del principio de resolución SLD donde la unificación clásica ha sido sustituida por un algoritmo de unificación borrosa basado en relaciones de proximidad. Por lo tanto, el algoritmo de unificación débil no falla si hay un encuentro de dos símbolos sintácticamente distintos siempre que sean aproximados, es decir, obtiene éxito con un cierto grado de aproximación. Por lo tanto, Bousi~Prolog calcula respuestas como grados de aproximación haciendo una clara distinción entre el conocimiento preciso y el vago. El conocimiento preciso en BPL es especificado por un conjunto de hechos y reglas Prolog, mientras que el conocimiento vago es principalmente especificado por un conjunto de lo que llamamos ecuación de proximidad, las cuales definen relaciones difuso-binarias que expresan cuán cerca están dos conceptos. Como ejemplo, un fragmento de una base de datos que almacena información sobre las personas. Supongamos algunos subconjuntos difusos sobre el dominio edad, de la que se han obtenido los grados de proximidad entre las etiquetas lingüísticas joven, adulto y viejo. Este conocimiento puede ser codificado por un conjunto de ecuaciones de proximidad, como lo muestra el siguiente fragmento de una base de datos deductiva.

```
% HECHOS Y REGLAS (Conocimiento preciso)

edad(mary, adulto).

edad(sam, joven).

edad(john, viejo).

amigo(X,Y):-edad(X,Z), edad(Y,Z), X \= Y.

% ECUACIONES DE PROXIMIDAD (CONOCIMIENTO VAGO O IMPRECISO)

joven ~ adulto = 0.75. viejo ~ joven = 0.25. adulto ~ viejo = 0.75.
```

En un sistema Prolog normal, si preguntamos sobre si Mary es amiga de Sam, "?-amigo (mary, sam)", el sistema falla, no obstante BPL nos permite obtener la respuesta "Sí con 0.75". Es decir, la verdad o falsedad sobre un hecho no es absoluta, más bien tiene algún grado de certeza.

Variables lingüísticas

Una variable lingüística (Zadeh, 1975) es una quíntupla áX, T(X), U, Gñ, donde: X es el nombre de la variable, T (X) es el conjun-

to de términos lingüísticos de X (es decir, el conjunto de nombres de valores lingüísticos o etiquetas lingüísticas de X), U es el dominio o universo de discurso, G es una gramática que permite generar T(X) y M es una regla semántica que asigna a cada término lingüístico x en T(X) su significado (es decir, un subconjunto difuso de U caracterizado por su función de pertenencia mx). Lo habitual es hacer distinción entre términos atómicos (términos primarios) y términos compuestos que se componen de términos primarios. Los subconjuntos borrosos que M aplica a los términos compuestos se calculan, mientras que los que se aplican a los términos primarios son definidos (en un manera subjetiva y dependiente del contexto). En Bousi~Prolog, para una variable X, sólo el dominio U y los subconjuntos borrosos asociados a los términos lingüísticos primarios en T(X) se consideran por su definición, el resto de términos compuestos se calculan automáticamente. Por otra parte, no hace una distinción léxica entre los componentes sintácticos y semánticos de X. Por lo tanto, hace uso de dos directivas para definir y declarar la estructura de una variable lingüística X. La directiva de dominio permite declarar y definir el dominio asociado a una variable lingüística. La sintaxis de esta directiva es:

":-domain(Nombre_Dominio (n,m,Magnitud).)", donde, Nombre_Dominio es el nombre del dominio, n y m (con n < m) son los límites inferior y superior del sub intervalo real [n, m], y la magnitud es el nombre de la unidad en la que se miden los elementos del dominio. La directiva fuzzy_set permite declarar y definir una lista de subconjuntos difusos (Los que se asocian a los términos principales de una variable lingüística) en un dominio predefinido. La sintaxis de esta Directiva es: ":-fuzzy set(Nombre_Dominio,[SubS_1(a1,b1,c1[,d1]),... ,SubS_n(an,bn,cn[,dn]]))."

Los Subconjuntos borrosos son definidos indicando su nombre, subs_i, y el tipo de función de pertenencia (funciones trapezoidales, si se dan cuatro argumentos, o funciones triangulares, si se utilizan tres argumentos).

Consultas lingüísticas difusas sobre información web

Pueden haber dos tipos de consultas flexibles: consultas lingüísticas difusas y consultas flexibles basadas en proximidad.

Primeramente emplearemos técnicas de harvesting para la extracción de información y el estándar XML para representar la información extraída. Posteriormente transformaremos cada registro del documento XML en predicados BPL mediante una herramienta construida para tal efecto. Luego se modela la imprecisión o conocimiento vago utilizando conjuntos borrosos y ecuaciones de proximidad para finalmente efectuar consultas lingüísticas difusas.

Extracción de la información

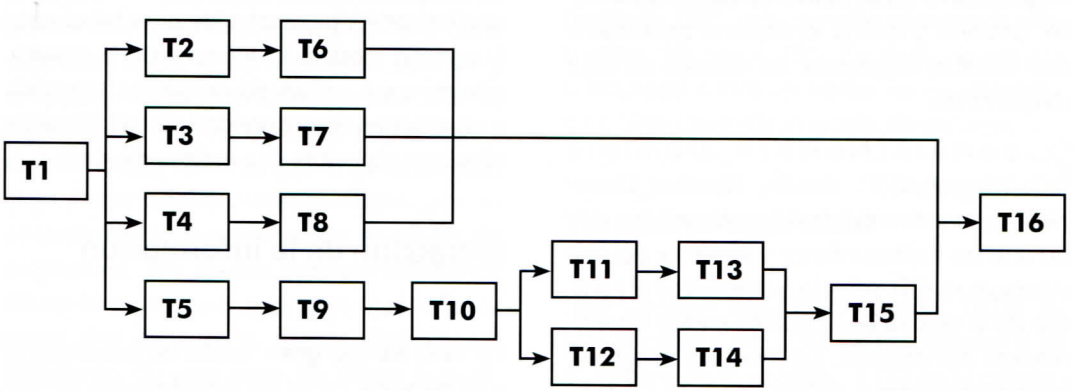
La Web es una gran fuente de información y el principal canal de transferencia de co-

nocimiento, por lo tanto, "es un buen lugar de donde extraer información". Si hablamos de extracción de información desde la web necesariamente se debe hacer mención del harvesting (del inglés cosechar) (Gatterbauer) como técnica de extracción (Handoko, 2008). Esta técnica se usa principalmente con contenido semi-estructurado (Rodríguez, 2004), es decir, listados, directorios, clasificados, toda aquella información en donde sea posible identificar patrones de cadenas de texto. Esta técnica se basa en el análisis del código HTML. Esta tarea requiere la programación de *robots* o *harvester* (Glez-Peña, Méndez, & Fdez-Riverola, 2007) que se encarguen de dicho análisis. Los robots pueden ser construidos utilizando diversas APIs existentes como Wget (GNU Operating System), Web Harvest (Nikic & Wajda), Crawler4j (Crawler4j) y jARVEST (Glez-Peña, Méndez, & Fdez-Riverola, 2007) (Daniel Glez-Peña) entre otros, no obstante esta implementación se realizó con jARVEST.

El robot encargado de la extracción se compone de una serie de piezas atómicas denominadas transformadores los cuales realizan una tarea específica. Cada trasformador recibe una entrada con la cual produce una salida. Tanto la entrada como la salida son cadenas de texto. Las operaciones de los transformadores pueden ser desde trasformación de una URL a su contenido HTML hasta búsquedas de patrones en el código usando XPATH (Berglund, y otros, 2011) o expresiones regulares (Watt). Para componer el robot final los transformadores se conectan en serie o en paralelo (Glez-Peña, Méndez, & Fdez-Riverola, 2007) como se muestra en la figura 2. Finalmente el robot es una serie de transformaciones para con una entrada de texto.

El ejemplo de la figura 1 corresponde a un extracto de un documento XML construido con los datos extraídos con el robot de la figura 2. El trasformador T1 (Wget) recibe como entrada una URL y entrega el conte-

FIGURA 2. CONEXIÓN SERIE/PARALELO DE TRANSFORMADORES



nido del documento HTML correspondiente, el cual se pasa como entrada a los trasformadores T2, T3, T4 y T5, los que mediante una consulta XPATH obtienen elementos de texto específicos como marca, año y precio del vehículo. La consulta XPATH que realiza T5 obtiene una nueva URL (formateada en T9) que es la entrada de otro Wget (T10) el cual entrega como salida un nuevo contenido HTML del cual se extraen dos datos más (color (T11) y cilindrada (T12)), los cuales se formatean como elemento XML. Finalmente la salidas de los transformadores conectaos en paralelo se mezclan primeramente en T15 y luego en T16. La salida final es un documento XML con muchos registros <vehículo></vehículo> como los del ejemplo 1. Para ver con detalle la utilización de la API jARVEST ver (Glez-Peña, Méndez, & Fdez-Riverola, 2007).

Transformación de la información

Una vez se tiene el documento XML, el paso siguiente es seleccionar las variables lingüísticas que serán objeto de consultas y transformar cada registro (tupla) en un predicado BPL equivalente. La primera tarea se realiza de manera asistida en una interface gráfica de la aplicación JAVA "conversor de XML a predicados" como se indica en la figura 3. Una vez que se ha seleccionando el o los atributos, la aplicación continúa sin intervención con la trasformación de las tuplas en predicados BPL.

En la Figura 3 se observa cómo luego de buscar y cargar un documento XML se muestra una lista con los atributos disponibles (Lista izquierda), de la cual son seleccionados los atributos precio, anio y color para generar los predicados (o lista) BPL sobre los cuales se realizarán las consultas flexibles. Al presionar el botón Transformar los predicados serán generados y guardados en un archivo de nombre nombreArchivo.xml.bpl. Luego de un mensaje confirmando el éxito de la operación, para el ejemplo de la figura 3 se generará el archivo xmlAnidado2-paper.xml.bpl. Dado que el documento XML contiene un solo registro, se generará el predicado "vehículo(anio#2013,precio#12600000,color('Rojo'))."

De haber seleccionado todos los atributos el predicado generado sería:

"vehiculo(marca('Toyota),anio#2013,precio#12600000,motor(cc#1200,tipo('En línea')),color('Rojo'))."

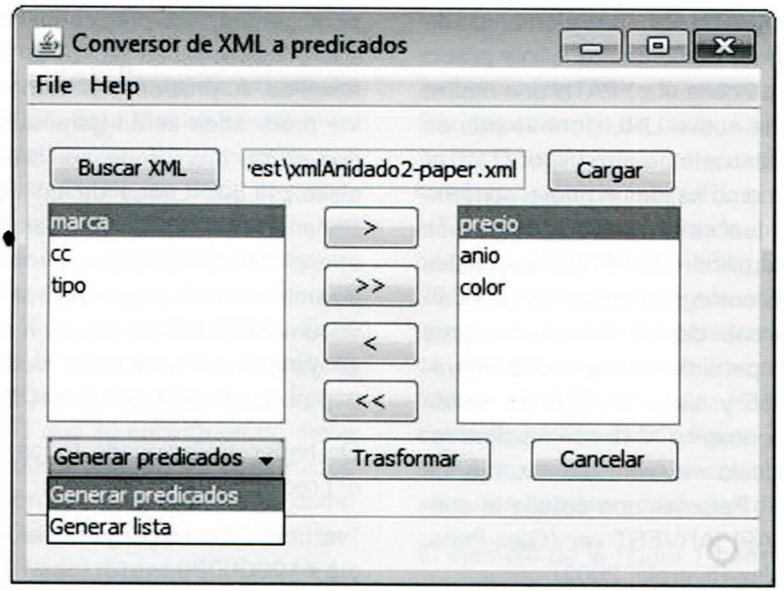
Una vez generados los predicados estos deben ser cargados en Bousi~Prolog para realizar las consultas lingüísticas.

Modelamiento de la imprecisión

Antes de poder efectuar consultas lingüísticas se requiere modelar la impresión. Bousi~Prolog permite efectuar dos tipos de consultas lingüísticas: consultas lingüísticas difusas y consultas flexibles basadas en proximidad.

a) Consultas lingüísticas difusas: Se realizan sobre datos numéricos como edad, peso, estatura, cilindrada de un vehículo,

FIGURA 3. CONVERSOR XML A PREDICADOS BOUSI~PROLOG



etc. Para realizar la consulta es necesario construir conjuntos borrosos, los que se asocian a términos primarios de una variable lingüística. Los conjuntos deben construirse sobre un domino definido previamente. Por ejemplo, para la variable lingüística Edad es posible definir los términos primarios joven, adulto y viejo con los conjuntos borrosos (0,0,20,30), (20,40,55,70) y (60,80,100,100) respectivamente definidos sobre el dominio edad cuyos límites inferior y superior se definen arbitrariamente como 0 y 100 respectivamente y años como la unidad de medida (Julián-Iranzo, Rubio-Manzano, & Gallardo-Casero, 2010). La sintaxis BPL tanto para definir el domino como los conjuntos borrosos es: ":-domain(Domain_name(n,m,Magnitude)).", donde, Domain_name es el nombre del dominio, n y m (con n < m) son los límites inferior y superior del subintervalo de los números reales [n,m], y Magnitude es el nombre de la magnitud en la que se miden los elementos del dominio. La sintaxis para definir los conjuntos borrosos es ":-fuzzy_set(Domain_name,[Subc_1(a1,b1,c1[,d1]),...,Subc_N(an,bn,cn[,dn])]).", donde Domain_name es el nombre del dominio definido previamente, Subc_1 es el nombre del conjunto borroso (términos primarios) más 3 o 4 argumentos numéricos para indicar si el tipo de función de pertenencia es triangular o trapezoidal. Para el

ejemplo de la variable lingüística Edad las instrucciones BPL serían las siguientes:

:-domain(edad(0,100,años)).

:-fuzzy_set(edad,[joven(0,0,20,30), adulto (20,40,55,70), viejo(60,80,100,100)]).

Después de esto será posible realizar consultas flexibles como ¿Quienes son jóvenes? mediante la sentencia "?-person(X,joven).".

b) Consultas flexibles basadas en proximidad: Este tipo de consultas en cambio, se realizan sobre datos no numéricos como rojo, sedan, station, Santiago, etc. Una relación de proximidad se denota por la ecuación a ~ b = a, la cual representa una entrada de una relación binaria difusa, donde a y b son símbolos de predicados y a es el grado de proximidad. Si dice entonces que a y b se relacionan en un grado a (Melton., 1999) (Julián Iranzo & Rubio Manzano, 2010). De esta manera, si disponemos de registros de vehículos almacenados conteniendo entre sus atributos color, podemos establecer las siguientes ecuaciones de proximidad: (rojo ~ negro = 0.6.), (rojo ~ amarillo= 0.4.), (amarillo ~ celeste = 0.8.) y (blanco ~ negro = 0.1.). Con esto podemos observar como el grado "0.1" de alguna manera está estableciendo que los colores blanco y negro se encuentran más distantes que los colores amarillo y celeste, para los cuales se estableció una relación de proximidad con grado "0.8", es decir, estos últimos colores se encuentran más próximos. Cualquiera sea el tipo de consulta, finalmente todo se traduce en términos de compilación a relaciones de proximidad, pues la semántica operacional de Bousi~Prolog es una adaptación del principio de resolución donde la unificación clásica se reemplaza por un algoritmo de unificación borrosa basado en relaciones de proximidad.

Conclusiones y Trabajo Futuro

Sobre los temas planteados podemos hacer varias reflexiones. Una de las mas importante probablemente se relaciona con la gran cantidad de información existente en la web, esto es una realidad, de ahí que se haga cada vez más frecuente hablar de "La información: divisa del futuro" o "El nuevo petróleo". Esto quiere decir que cada vez mayor es el valor que adquiere la información, pues con ella se puede apoyar los procesos de toma de decisiones. A mayor información menor incertidumbre, lo que se traduce en decisiones tomadas con una alta probabilidad de acierto y esto, claramente se puede cuantificar en un ahorro deducible de la reducción de costos a nivel de gestión. Por otra parte, la abundante información inconexa existente en la web puede, mediante los procesos apropiados, producir nueva información con valor agregado, la cual hoy en día es perfectamente comercializable. Son muchas las empresas que se dedican exitosamente al procesamiento de datos y a construir información para ser comercializada a empresas que por ejemplo la emplean en estudios de mercado o análisis de prospectos. Nos referimos a verdadera información construida, no listas de correo ni base de datos de contactos, sino información del tipo "¿Qué

marca de celular se prefiere?, ¿Cuáles prefieren los hombres o las mujeres?", "¿Qué color de vehículo se vende más o menos, etc.?" , lo maravilloso de esto, es que materia prima existe en abundancia en la web y lo mejor, es gratis. No obstante, la única dificultad de momento es que los mecanismos para un uso medianamente eficiente no se encuentran muy difundidos, y en este sentido, este trabajo pretende acercar un poco las posibilidades y promesas del webminig, en donde hemos tratado de abstraer al máximo la complejidad subyacente para presentar un enfoque atractivo que permita tanto la obtención de información desde la web como la posibilidad de realizar consultas sobre ella. Lo particular de este enfoque es que permite incorporar flexibilidad en las consultas mediante la utilización de términos lingüísticos. Esto implica que es necesario manejar y modelar la ambigüedad que reviste cada termino lingüístico (alto, bajo, chico, barato, joven, etc), lo cual demanda la existencia de conocimiento experto o experiencia que debe ser plasmada en el modelo mediante conjuntos difusos o grados de proximidad. Por ejemplo, si hablamos de joven, para mí es una persona entre 15 y 30 años, pero para otra puede estar entre 17 y 37. Lo mismo sucede con el precio de un vehículo, ¿Cuánto es caro o barato?. Información como esta proviene necesariamente de la experiencia de las personas, lo cual no necesariamente es un inconveniente, pero es algo que debe tenerse en cuanta. Una forma simple de sintetizar esta experiencia o conocimiento es mediante un pequeño estudio estadístico sobre el dominio de información. De modo tal que sea posible conocer muestralmente, en un contexto estadístico, cuanto es caro, barato, joven, etc.

Nuestro planteamiento, si bien es cierto cumple con el objetivo de formar bases de conocimientos con información extraída desde la web para luego consultarla de manera tradicional o agregando flexibilidad en la consulta, no es una solución definitiva para un usuario común, pues reviste algunos tecnicismos y bastante laboriosidad que obliga, ante una implementación escalable, contar con competencias muy específicas sobre todo en materia de modelamiento y construcción de los agentes de software encargados de la extracción, pues estos tienen una fuerte dependencia con la estructura de los sitios web. Lo mismo sucede con el modelamiento de la imprecisión para dotar de flexibilidad a las consultas.

Si bien es cierto esta complejidad no puede ser eliminada de momento, como trabajo futuro, resulta tremendamente necesario encapsularla al menos o proveer mecanismos de asistencia mediante una capa adicional de abstracción donde operen interfaces de software que asistan en las tareas de generación de los robots y modelamiento de la imprecisión. A pesar de lo señalado, es una propuesta que posibilita obtener resultados en el muy corto plazo, prácticamente de forma inmediata, en comparación con otras alternativas en el contexto de la web semántica donde se requieren cambios profundos en la web y por ende mayor tiempos para obtener resultados en materia de utilización de información web para apoyar los procesos de toma de decisiones.

Bibliografía

- **Al**, M. K. (1990). Fuzzy Query using Clustering techniques Information Processing and Management. *26* (2), 279-293.

- **Berglund**, A., **Boag**, S., **Chamberlin**, D., **Fernández**, M. F., **Kay**, M., **Robie**, J., y otros. (2011). *XML Path Language (XPath) 2.0 (Second Edition).* Recuperado el 22 de 5 de 2013, de W3C Recommendation 14 December 2010 (Link errors corrected 3 January 2011): http://www.w3.org/TR/2010/REC-xpath20-20101214/

- **Crawler4j**. (s.f.). Recuperado el 21 de 5 de 2013, de https://code.google.com/p/crawler4j/

- **Date**, C., & Darwen, H. (1997). *A guide to the SQL standard: A user's guide to the standard database language SQL.* Addison-Wesley.

- **Eisenberg**, A., & **Melton**, J. (s.f.). SQL/XML and the SQLX Informal Group of Companies.

- **Gatterbauer**, W. (s.f.). Web Harvesting.

- **Glez-Peña**, D. ; **R.-R.** , J. (s.f.). *jARVEST (Java web harvesting library)* . Recuperado el 21 de 5 de 2013, de http://sing.ei.uvigo.es/jarvest/

- **Glez-Peña**, D., **Méndez**, J. R., & **Fdez-Riverola**, F. (2007). Automator: herramienta flexible para la extracción de información en sitios web bioinformáticos.

- **GNU** Operating System. (s.f.). *GNU Wget.* Recuperado el 21 de 5 de 2013, de http://www.gnu.org/software/wget/: http://www.gnu.org/software/wget/

- **Handoko**, Z. A. (2008). A Simple Mechanism for Focused Web-harvesting.

- **Hsiao**, S. C. (2007). A New Approach for Fuzzy Query Processing Based on Automatic Clustering Techniques. *Information and Management Sciences* , 223-240.

- **Julián Iranzo**, P., & **Rubio Manzano**, C. (2010). BOUSI~PROLOG: A Fuzzy Logic Programming Language for Modeling Vague Knowledge and Approximate Reasoning. *International Conference on Fuzzy Computation (ICFC 2010).*

- **Julián-Iranzo**, P., **Rubio-Manzano**, C., & **Gallardo-Casero**, J. (2010). Inclusión de Conjuntos Borrosos en el Núcleo del Sistema Bousi~Prolog. *XV Congreso Español Sobre Tecnologías y Lógica Fuzzy.* Huelva.

- **Kannan**, V. B. (2010). A Framework for Computing Linguistic Hedges in Fuzzy Queries. *The Int. J. of Database Management Systems , 2* (1).

- **Lee**, R. C. (1972). Fuzzy Logic and the Resolution Principle. *Journal of the ACM* , 119-129.

- **Lloyd**, J. W. (1987). Foundations of Logic Programming.

- **Melton**, S. S. (1999). Proximity relations in the fuzzy relational database model.

- **Nikic**, V., & **Wajda**, A. (s.f.). *Web Harvest.* Recuperado el 21 de 5 de 2013, de http://web-harvest.sourceforge.net/contact.php

- **Orchard**, R. (1998). FuzzyClips Version 6.04A. User's Guide. *Integrated Reasoning.Institute for Information Technology.*

- **P. Julián**, C. R. (2009). Bousi~Prolog: a Prolog Extension Language for Flexible Query Answering. *Electronic Notes in Theoretical Computer Science* , 131-147.

- **Pankowski**, T. (202). *XML-SQL: An XML Query Language Based on SQL and Path Tables.* Springer Berlin Heidelberg.

- **Petry**, B. B. (1985). A fuzzy model for relational databases. *Fuzzy Sets and Syst.* , 213-226.

- **Rodríguez**, J. A. (2004). *La estructura de los documentos en el ámbito de recuperación de información: propuestas para su compresión, indexación y recuperación.* Valladolid, España.

- **Rubio**, P. J. (2009). A similarity-based WAM for Bousi~Prolog. *In Proceedings of IWANN 2009, Springer LNCS 5517* , 245-252.

- **Sessa**, M. I. (2002). Approximate reasoning by similarity-based SLD resolution. *Theoretical Computer Science* , 389-426.

- **Tahami**, V. (1977). A conceptual framework for fuzzy query processing - a step toward very intelligent databases systems. Information Processing and Management. *13*.

- **Takahashi**, Y. (1995). A fuzzy query language for relational databases. *Fuzziness in Database Management Systems* .

- **W3C** (s.f.). *Extensible Markup Language (XML) 1.0 (Fifth Edition).* Recuperado el 23 de 05 de 2013, de W3C: *http://www.w3.org/TR/REC-xml/*

- **Watt**, A. *Beginning Regular Expressions.* Indianapolis, India: Wiley Publishing, Inc.

- **Yoshikawa**, M., & **Amagasa**, T. (2001). XRel: A Path-Based Approach to Storage and Retrieval of XML Documents Using Relational Databases. *1* (1).

- **Zadeh**, L. A. (1975). The Concept of a Linguistic Variable and its Applications to Approximate Reasoning I, II and III. *J. of Information Sciences 8 and 9, Elsevier* .

- **Zadeh**, L. A. (1965). Fuzzy Sets.Information and Control.