ISSN 1870-4069

Interfaz de lenguaje natural para consultar cubos multidimensionales utilizando procesamiento analítico en línea

J. A. Porras Medrano, R. Florencia-Juárez, G. Rivera Zárate, V. García Jiménez

Universidad Autónoma de Ciudad Juárez, Chihuahua, México

al133614@alumnos.uacj.mx,
{rogelio.florencia, gilberto.rivera, vicente.jimenez}@uacj.mx

Resumen. El Procesamiento Analítico en Línea (On-Line Analytical Processing, u OLAP, por sus siglas en inglés) es una solución en el área de Inteligencia de Negocios que emplea estructuras multidimensionales, es decir, cubos OLAP, con el fin de agilizar el procesamiento y el acceso a grandes volúmenes de datos, normalmente almacenados en un almacén de datos (DW). Para acceder a los datos almacenados en un cubo, en muchas ocasiones se requiere que los usuarios formulen consultas en el lenguaje MultiDimensional eXpressions (MDX), conocimiento técnico que la mayoría de los usuarios no posee. En este trabajo se describe la implementación de una Interfaz de Lenguaje Natural (ILN) cuyo objetivo es traducir una consulta formulada en lenguaje natural a una consulta MDX. El conocimiento que la ILN requiere para efectuar la traducción es modelado semánticamente a partir de los distintos elementos que componen la estructura de los cubos. El modelado es realizado automáticamente por un módulo de configuración, el cual utiliza representaciones semánticas diseñadas en el Lenguaje de Ontologías Web (Web Ontology Language, OWL por sus siglas en inglés).

Palabras clave: interfaz de lenguaje natural, conocimiento semántico, consulta MDX, cubos OLAP, base de datos multidimensionales.

Natural Language Interface for Querying Multidimensional Cubes by using On-Line Analytical Processing

Abstract. On-Line Analytical Processing (OLAP) is a solution in the Business Intelligence field that uses multidimensional structures, that is to say, OLAP cubes, with the aim of speed up processing and access to large volumes of data, usually stored in a data warehouse (DW). In order to access the data stored in a cube, it is very often for users to formulate queries in the MultiDimensional eXpressions language, but that is technical knowledge which most of them do not possess. This work describes the implementation of a Natural Language Interface (NLI) which main objective is to translate a query formulated in natural language to an MDX query. The knowledge required for the NLI to perform the translation process is modeled semantically from the different elements that

compose the structure of the cubes. Modeling is performed automatically by a configuration module, which uses semantic representations designed in the Ontologies Web Language (OWL).

Keywords: natural language interface, semantic knowledge, MDX query, OLAP cubes, multidimensional databases.

1. Introducción

El área de Inteligencia de Negocios está enfocada en transformar datos provenientes de sistemas de gestión empresarial en conocimiento, con lo cual se pueda optimizar el proceso de toma de decisiones estratégicas. Para lograr esto se han desarrollado numerosas aplicaciones y herramientas de software que permiten obtener información significativa de fuentes de información y datos, tales como bases de datos y almacenes de datos (DataWarehouse, DW por sus siglas en inglés).

En algunas ocasiones, realizar tal labor requiere normalmente de cierto grado de conocimientos técnicos de lenguajes de consultas como SQL (Structured Query Language) o MDX (MultiDimensional eXpressions), obligando a usuarios convencionales a recurrir a expertos con el fin de recuperar información de las fuentes mencionadas.

Para facilitar a los usuarios este proceso, investigadores se han enfocado en diseñar interfaces que permitan traducir automáticamente una consulta expresada en lenguaje natural por los usuarios a una consulta MDX.

En este trabajo se propone la arquitectura de una interfaz de lenguaje natural (ILN) capaz de traducir de una consulta formulada por el usuario en el idioma inglés a una consulta MDX, la cual se utiliza para extraer información de cubos multidimensionales, almacenados en un DW. El funcionamiento se divide principalmente en dos módulos, el Módulo Generador de Conocimiento (MGC) y el Módulo de la Interfaz (MI). El MGC se encarga de modelar semánticamente el conocimiento que el MI utiliza para interpretar la consulta del usuario y generar la consulta MDX correspondiente.

El modelado se genera automáticamente a partir de la estructura de los cubos de un DW, utilizando representaciones semánticas diseñadas en el Lenguaje de Ontologías Web (Web Ontology Language, u OWL, por sus siglas en inglés). En la Sección 2 se presentan trabajos relacionados; en la Sección 3 se presenta la arquitectura de la interfaz propuesta; en la Sección 4 se presenta una discusión acerca del desempeño de la interfaz y, en la Sección 5 se presentan conclusiones y trabajos futuros.

2. Trabajos relacionados

En el trabajo de Kuchmann-Beauger et al. [1] se propone una ILN capaz de procesar preguntas expresadas por los usuarios en lenguaje natural. Su enfoque consiste en identificar palabras clave o entidades en la expresión ingresada por el usuario y enlazar dichas entidades a objetos definidos previamente en un modelo de datos. Su objetivo fue demostrar que reescribir la consulta del usuario a través del uso de un tesauro cuando no se ha encontrado ninguna respuesta, produce mejores resultados. Los elementos semánticos identificados en la pregunta del usuario se traducen a una

consulta MDX que el sistema pueda entender. Su ILN contiene un grafo de objetos y restricciones descritos en el modelo del DW.

Si bien la propuesta muestra resultados prometedores, al mismo tiempo los autores reconocen que el sistema aún podría ser más preciso si se convierte el tesauro en una ontología que ya no solo permita reescribir la consulta del usuario, sino además recolectar información de fuentes no estructuradas para validar las respuestas proporcionadas.

Por otro lado, en el trabajo de Saias et al. [2] se desarrolló BINLI, una ILN enfocada al Procesamiento Analítico En Línea (OnLine Analytical Processing, OLAP por sus siglas en inglés) cuyo funcionamiento se basa en una ontología. Las herramientas OLAP son soluciones para análisis de datos multidimensionales que permiten al usuario controlar la perspectiva y grado de detalle en cada dimensión del análisis en grandes volúmenes de datos.

BINLI pretende simplificar y hacer más flexible e intuitiva la interacción de usuarios con herramientas OLAP permitiéndoles realizar consultas en NL. A diferencia del trabajo de Kuchmann-Beauger et al., cuya ILN tiene un tesauro como base de conocimiento, BINLI se encuentra basado en una ontología, permitiendo inferencia en procesos de complejidad semántica mayores. BINLI utiliza análisis gramatical, reconocimiento de entidades, análisis morfosintáctico, cálculo de similitud semántica y razonamiento semántico. Cuenta con un generador de consultas OLAP MDX que permite evaluar varias interpretaciones de una misma consulta ordenadas descendentemente por orden de peso. La interpretación de mayor peso es enviada a un motor OLAP para su ejecución.

Como motor OLAP utilizan Pentaho Analysis Services Community Edition. BINLI también detecta errores de escritura utilizando la distancia Levenshtein. Además, detecta relaciones indirectas probando la compatibilidad semántica entre los términos a través del uso de una ontología de apoyo.

La propuesta de Prat, Akoka at al. [3] presenta un enfoque que busca definir una ontología basada en lógica descriptiva (OWL-DL) a partir de un modelo multidimensional. Las dimensiones, jerarquías de las dimensiones, niveles de la dimensión de la jerarquía, atributos del nivel de la dimensión, rollups, hechos, mediciones, funciones de agregación y tipos de agregación del modelo dimensional son representados en una ontología OWL-DL. Con el uso de la ontología OWL-DL demostraron facilitar la inferencia de la información deseada por el usuario, además de permitir una representación concisa y formal del conocimiento.

Configurar una ILN para una base de datos es un trabajo considerable, por tal motivo, Popescu et al. [4] decidieron implementar una ILN llamada PRECISE con la inclusión de un parser (analizador gramatical) estadístico. Implementaron un modelo semántico robusto para corregir errores de análisis gramatical y un marco de trabajo teórico para diferencia entre preguntas manejables y difíciles. Entrenaron el parser utilizando un conjunto de 150 preguntas donde cada palabra es etiquetada con etiquetas *Parte del Discurso* (Part of Speech, PoS por sus siglas en inglés) [5]. Utilizaron el parser Charniak como base para sus experimentos. PRECISE puede corregir errores sintácticos, complementos de preposición y elipsis de preposición. Cuando una decisión del parser resulta inconsistente con la información semántica del léxico de PRECISE, este último intenta reparar el árbol de análisis gramatical del parser. No obstante, PRECISE no puede realizar este tipo de correcciones en ciertos problemas

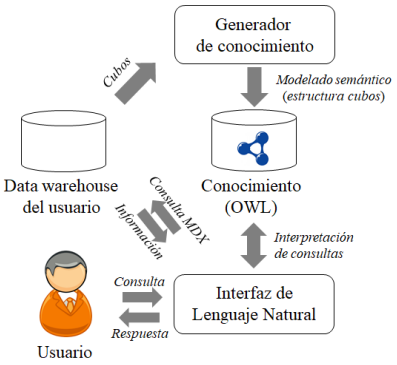

Fig. 1. Módulos de la interfaz.

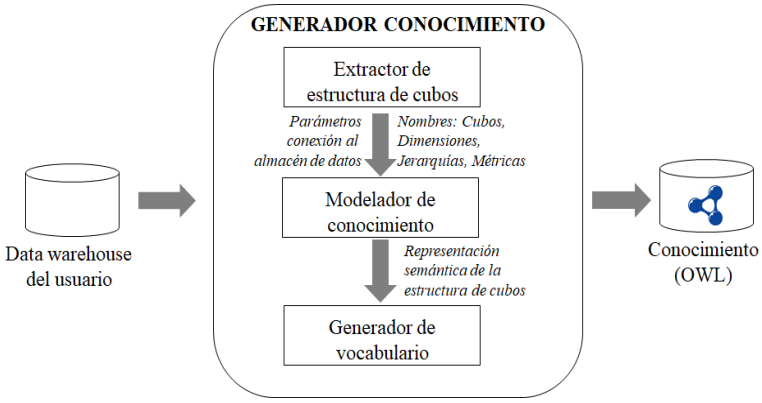

Fig. 2. Arquitectura del Módulo Generador de Conocimiento.

como acoplamiento de verbos, de cláusulas, frases numéricas de sustantivos, entre otros, obligando al usuario a reformular su pregunta.

3. Arquitectura propuesta

La ILN propuesta en este artículo requiere de conocimiento del dominio, es decir, requiere conocer la estructura de los cubos multidimensionales almacenados en el DW con el fin de interpretar las consultas en lenguaje natural y generar las consultas MDX correspondientes. Por tal motivo, el diseño de la ILN se dividió en dos módulos, el *Módulo Generador de Conocimiento* (MGC) y el *Módulo de la Interfaz* (MI). En la Fig. 1 se presentan ambos módulos. En la Sección 3.1 se describe el MGC y en la Sección 3.2 se describe el MI.

3.1. Módulo generador de conocimiento

Como se pudo ver en la Fig. 1, el objetivo del MGC es generar el conocimiento del MI. Para este fin, se analiza la estructura de cada uno de los cubos multidimensionales

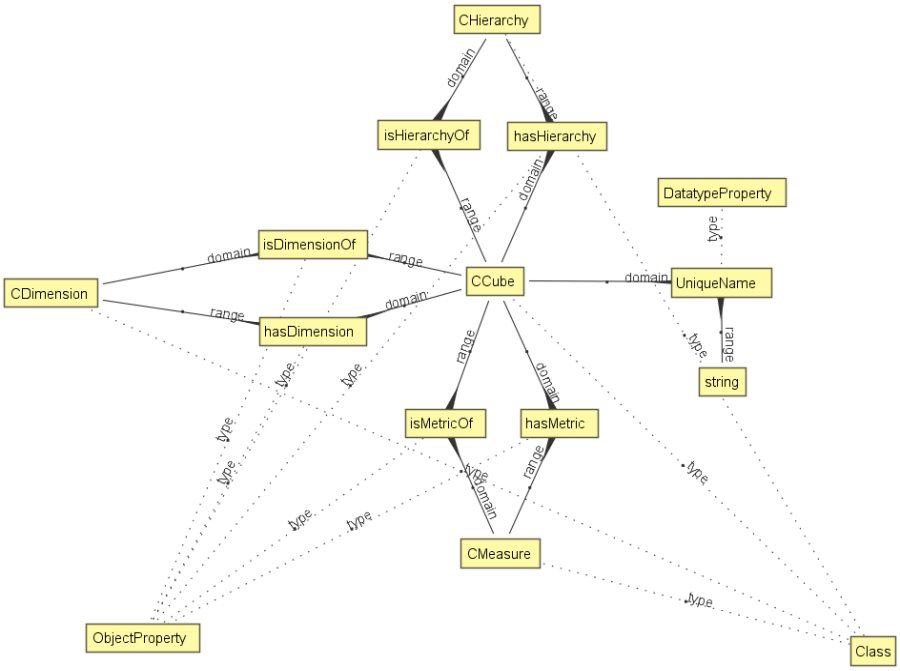

Fig. 3. Estructura de la representación semántica diseñada para modelar la estructura de los cubos.

almacenados en el DW con el objetivo de identificar los elementos que la componen (nombres de los cubos, dimensiones, jerarquía y métricas). Posteriormente, los elementos identificados son modelados semánticamente de manera automática utilizando una representación semántica que fue diseñada para este propósito.

Por último, el modelado semántico es almacenado en una ontología, es decir, un archivo con formato XML y extensión OWL el cual constituye el conocimiento del MI. En la Fig. 2 se puede observar la arquitectura del MGC.

Como se puede apreciar en la Fig. 2, el MGC recibe como entrada los parámetros de conexión al DW. Mediante estos parámetros se establece una conexión con el DW y el *Extractor de estructura de cubos* analiza cada uno de los cubos del DW y extrae sus nombres, dimensiones, jerarquías y métricas. Cabe mencionar que sólo se extraen elementos que conforman la estructura de un cubo y no los datos que éste almacena. Los elementos extraídos son enviados al *Modelador de conocimiento*, el cual utiliza una representación semántica que fue diseñada para representar estos elementos.

La estructura de la representación semántica diseñada consta de clases (*CCube*, *CDimension*, *CHierarchy*, *CMeasure*), propiedades de objeto (*hasDimension*, *isDimensionOf*, *hasHierarchy*, *isHierarchyOf*, *hasMeasure*, *isMeasureOf*) y propiedades de datos (*UniqueName* de tipo *string*). Cada nombre de cubo es modelado como un individuo instanciado a partir de la clase *CCube*. Cada nombre de las dimensiones de un cubo es modelado como un individuo de la clase *CDimension* y es relacionado al individuo *CCube* al que pertenece mediante las propiedades de objeto *isDimensionOf* y *hasDimension* y así sucesivamente con los demás elementos

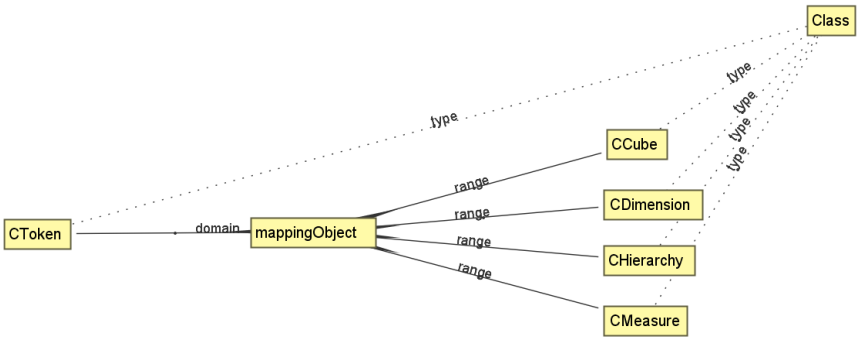

Fig. 4. Estructura de las representaciones semánticas diseñadas para modelar el vocabulario y su relación con los elementos del modelado semántico.

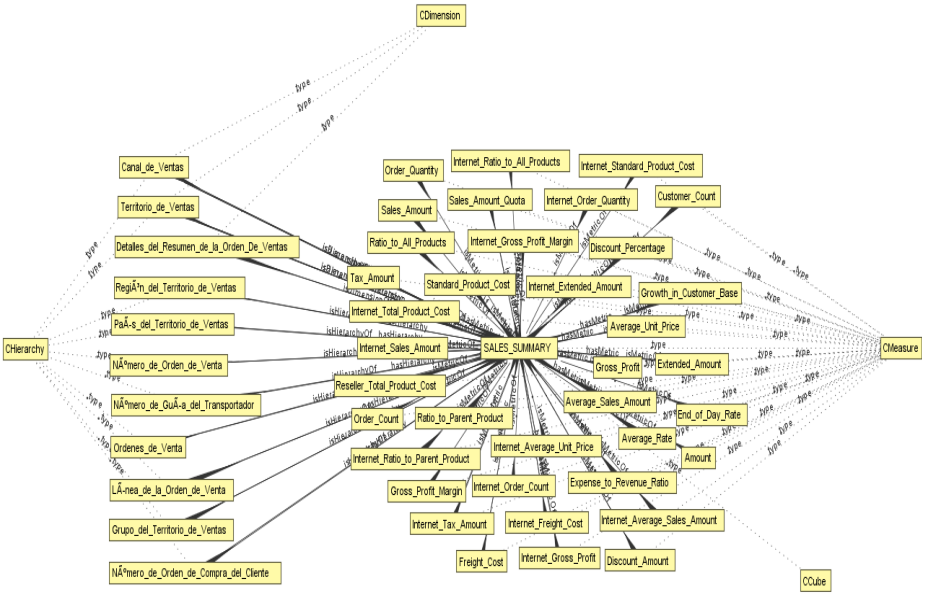

Fig. 5. Ejemplo del conocimiento modelado y almacenado en una de las ontologías generadas.

identificados. En la Fig. 3 se presenta la estructura de la representación semántica diseñada.

Después de haber modelado semánticamente la estructura de los cubos, el modelado es enviado al *Generador de vocabulario* cuyo objetivo es construir el vocabulario inicial del MI, el cual se compone de todos los nombres de cada uno de los elementos modelados.

Posteriormente, para incrementar la cobertura lingüística, cada palabra en el vocabulario es adicionada con sinónimos, así como con palabras que compartan el mismo lema.

Finalmente, todas las palabras son agregadas al modelado semántico utilizando una representación semántica diseñada para este fin. Esta representación consta de una clase llamada *CToken* y de las propiedades de objeto *mappingObject* e *isMappedBy*. Cada palabra es modelada como un individuo de la clase *CToken* y este individuo es relacionado a través de la propiedad *mappingObject* a los elementos a los que éste mapea, es decir, a los elementos de los que fue generado.

Es importante mencionar que este proceso es esencial para que el MI pueda interpretar las consultas de los usuarios. En la Fig. 4 se puede observar la estructura de la representación semántica diseñada para modelar el vocabulario y su relación con los elementos del modelado semántico.

Posterior a los procesos anteriormente mencionados, se genera la ontología (archivo de extensión.owl) en la que se va a almacenar el modelado semántico generado por el MGC.

En la Fig. 5 se presenta un fragmento del conocimiento modelado a partir del cubo SALES SUMMARY de la base de datos AdventureWorksDW2014, la cual es una base de datos de ejemplo de Microsoft. En la Fig. 5 se pueden ver las clases *CCube*, *CDimension*, *CHierarchy* y *CMeasure*, las cuales identifican las dimensiones, las jerarquías, las métricas y el nombre del cubo.

Los nombres mostrados son instancias de estas clases. Cabe mencionar que para facilitar la interpretación de la Fig. 5, no se muestran los respectivos sinónimos y palabras que comparten el mismo lema que los nombres de las instancias de estas clases.

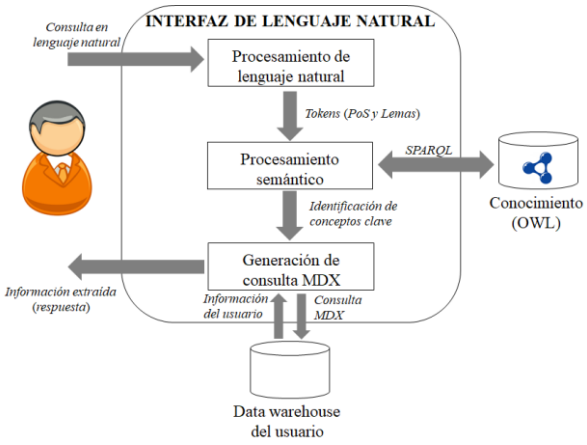

Fig. 6. Arquitectura del Módulo de la Interfaz.

3.2. Módulo de la interfaz

El objetivo del MI es traducir la consulta formulada en lenguaje natural por el usuario a una consulta MDX. Como primer paso se realiza un *procesamiento de lenguaje natural* a la consulta introducida por el usuario. Posteriormente se realiza un

Tabla 1. Procesamiento de lenguaje natural realizado por la herramienta Freeling.

TOKEN	LEMA	POS	SIGNIFICADO POS
Show	show	VB	pos=verb vform=infinitive
Me	me	PRP	pos=pronoun type=personal
Internet	internet	NP	pos=noun type=proper
sales	sales	NNS	pos=noun type=common num=plural
amount	amount	NN	pos=noun type=common num=singular
As	as	IN	pos=preposition
Per	per	IN	pos=preposition
customer	customer	NN	pos=noun type=common num=singular
.	.	FP	pos=punctuation type=period

procesamiento semántico con el fin de identificar sin ambigüedad conceptos clave en la consulta, así como los elementos de la ontología a los que hacen referencia. Finalmente, en base a los elementos identificados, se realiza el proceso de *generación de la consulta MDX*. En la Fig. 6 se puede apreciar la arquitectura del MI.

3.3. Procesamiento de lenguaje natural

Este paso consiste en analizar la consulta introducida por el usuario utilizando Freeling [6], la cual es una suite de herramientas utilizadas en el procesamiento de lenguaje natural que ofrece soporte para diversos idiomas, entre ellos, inglés y español.

La herramienta Freeling es utilizada para: *a*) separar la consulta del usuario en tokens, *b*) etiquetar gramaticalmente cada token de acuerdo a su función dentro de la consulta (Part of Speech) y *c*) obtener los lemas de cada token. Por ejemplo, si el usuario introduce la consulta "*Show me Internet sales amount as per customer."* en el idioma inglés, ésta es separada en tokens como se muestra a continuación:

"*Show*" | "*me*" | "*Internet*" | "*sales*" | "*amount*" | "*as*" | "*per*" | "*customer*" | "."

En seguida, se etiqueta gramaticalmente cada uno de los tokens y se determinan sus lemas. En la Tabla 1 se muestra el resultado del procesamiento realizado por Freeling. La información gramatical es asignada a sus respectivos tokens en la memoria del MI para ser enviada a las siguientes fases de procesamiento. Cada una de las siguientes fases actualizará la información de los tokens.

Tabla 2. Ejemplo de mapeos asignados a tokens.

TOKEN	LEMA	POS	SIGNIFICADO POS
Show	Show	VB	
me	me	PRP	
Internet	internet	NP	[Internet Sales Order Details].[Sales Order Line].[Sales Order Line] **[Measures].[Internet Average Sales Amount]** **[Measures].[Internet Sales Amount]** [Measures].[Internet Tax Amount] [Measures].[Internet Total Product Cost]
Sales	sales	NNS	[Employee].[Sales Person Flag].[Sales Person Flag] [Internet Sales Order Details].[Sales Order Line].[Sales Order Line] **[Measures].[Internet Average Sales Amount]** **[Measures].[Internet Sales Amount]** [Measures].[Reseller Average Sales Amount] [Sales Channel].[Sales Channel].[Sales Channel] [Sales Reason].[Sales Reason].[Sales Reason] [Sales Summary Order Details].[Sales Orders].[Sales Orders]
amount	amount	NN	[Measures].[Reseller Average Sales Amount] **[Measures].[Internet Average Sales Amount]** [Measures].[Internet Extended Amount] **[Measures].[Internet Sales Amount]** [Measures].[Internet Tax Amount] [Measures].[Reseller Sales Amount] [Measures].[Reseller Tax Amount] [Measures].[Sales Amount Quota]
as	as	IN	
per	per	IN	
customer	customer	NN	[Customer].[Customer].[Customer] [Clustered Customers].[Customer Clusters].[Customer Clusters] [Customer].[City].[City] [Measures].[Customer Count] [Measures].[Growth in Customer Base]
.	.	FP	

3.4. Procesamiento semántico

El objetivo de esta fase es identificar sin ambigüedad conceptos clave en la consulta del usuario. Un concepto clave es aquel token o conjunto de tokens (token combinado) que mapean a elementos de la ontología.

Para identificar los conceptos clave se recorre cada uno de los tokens etiquetados gramaticalmente como sustantivo, adjetivo o verbo. Por cada token, el MI genera una serie de consultas SPARQL [7] con el fin de acceder al conocimiento modelado en la

Tabla 3. Ejemplo de un token compuesto.

Token	Lema	PoS	Significado PoS
Show	Show	VB	
me	me	PRP	
Internet sales amount	Internet sales amount	NP	[Measures].[Internet Average Sales Amount] [Measures].[Internet Sales Amount]
as	as	IN	
per	per	IN	
customer	customer	NN	[Customer].[Customer].[Customer] [Customer].[City].[City] [Measures].[Customer Count] [Measures].[Growth in Customer Base]
.	.	FP	

ontología para identificar si el token coincide textualmente con algún individuo de la clase *CToken*.

Si existe alguna relación token – *CToken*, el MI genera nuevamente una serie de consultas SPARQL para identificar, a través de la propiedad de objeto *mappingObject*, los elementos de la ontología (*CCube*, *CDimension*, *CHierarchy* y/o *CMeasure*) mapeados por el individuo *CToken*. Todos los elementos mapeados son asignados en memoria al token correspondiente. En la Tabla 2 se presentan sólo algunos ejemplos de mapeos asignados a tokens, ya que la base de datos AdventureWorksDW2014 almacena varios cubos multidimensionales compuestos por varias dimensiones, jerarquías y métricas.

Posteriormente, el MI busca reducir el número de tokens de la consulta. Para este fin, se analizan los mapeos asignados a cada token y se aplican algunas reglas gramaticales sencillas, definidas a priori. Los tokens que tienen asignados al menos un mapeo en común y que satisfacen las reglas gramaticales son combinados para formar un solo token compuesto, conservando los mapeos en común y eliminando los mapeos diferentes. Al reducir el número de tokens y al reducir sus mapeos, se simplifica el proceso de generación de la consulta MDX. Por ejemplo, las reglas gramaticales indican al MI que puede intentar combinar los tokens *Internet*, *sales* y *amount*, pero no *customer*. A continuación, el MI analiza los mapeos de estos tokens y determina que tienen en común los mapeos: [Measures].[Internet Average Sales Amount] y [Measures].[Internet Sales Amount]. Por tal motivo, el MI los combina en uno solo token y descarta los mapeos restantes.

Adicionalmente, el MI determina que ambos mapeos pertenecen al mismo cubo, *Adventure_Works*. Con respecto a *customer*, los mapeos que no pertenecen al cubo *Adventure_Works* son descartados. En la Tabla 3 se puede observar que *Internet sales amount* ahora es un token compuesto, el cual contiene solamente los mapeos [Measures].[Internet Average Sales Amount] y [Measures].[Internet Sales Amount]. Adicionalmente, también se puede notar que el mapeo [Clustered Customers].[Customer Clusters].[Customer Clusters] de *customer* fue descartado debido a que pertenece al cubo *Mined_Customers*.

Por último, el MI identifica los tokens ambiguos con el propósito de mostrar un diálogo de desambiguación que permita al usuario especificar el contexto. Un token que poseen mapeos a diferentes elementos, es decir que mapean a diferentes dimensiones, jerarquías o métricas es considerado como ambiguo.

3.5. Generación de consulta MDX

Después de que el MI logró identificar sin ambigüedad los tokens y sus respectivos mapeos en los procesos descritos en las secciones anteriores, se procede a generar la consulta MDX.

De manera general, para generar una consulta MDX, se analiza cada uno de los tokens que poseen mapeos asignados. Es importante mencionar que, para generar la consulta, se verifica que todos los mapeados de los tokens pertenezcan a un mismo cubo. Esto es debido a que, hasta nuestro conocimiento, no se puede acceder a más de un cubo en una sola consulta MDX.

Todos los elementos que mapean a individuos de la clase *CMeasure*, son agregados a la cláusula SELECT. Posteriormente, se incluye la cláusula ON COLUMNS.

Todos los elementos que mapean a individuos que no pertenecen a la clase *CMeasure*, son agregados a la cláusula SELECT. Posteriormente se incluye la cláusula ON ROWS. Es importante mencionar que, si estos elementos son de diferentes dimensiones del cubo, se incluye la cláusula CROSSJOIN.

La consulta MDX generada a partir de la consulta utilizada como ejemplo en la Sección 3.2 se presenta a continuación:

```
SELECT {[Measures].[Internet Sales Amount]} on Columns,
        {[Customer].[Customer].[Customer]} on Rows
FROM [Adventure Works]
```

4. Discusiones

A pesar de que el desarrollo de ILNs tuvo sus inicios en los años 60, actualmente no son capaces de responder correctamente todas las preguntas formuladas por los usuarios y, en consecuencia, existen muchos problemas que aún continúan abiertos a la investigación.

Sin lugar a duda, el lenguaje natural representa por sí mismo uno de los principales problemas a vencer debido a su riqueza lingüística y a diversas situaciones que en este se presentan, tales como ambigüedades (léxicas, sintácticas, semánticas, etc.), elipsis (nominal, verbal, preposición, etc.), anáforas (nominal, pronominal, etc.), entre otros. Es del principal interés de la comunidad científica hacer frente a estas situaciones, ya que limitan la capacidad de entendimiento de una interfaz, afectando negativamente su desempeño. Adicionalmente, entre los problemas inherentes a una interfaz se encuentran los relacionados con el proceso de traducción, con la portabilidad a diferentes bases de datos y con la forma de representar el conocimiento, entre otros. Abordar satisfactoriamente la mayoría de estos problemas depende del conocimiento del que disponga la interfaz.

Con el desarrollo de este trabajo se intenta explorar qué elementos deben conformar el conocimiento de una interfaz, cómo obtenerlos de una manera automatizada para

facilitar la portabilidad de la interfaz y cómo modelarlos. Con este trabajo se pretende construir una base sólida que permita abordar problemas complejos relacionados con el lenguaje natural, como los mencionados anteriormente.

Hasta el momento, el conocimiento generado por el MGC descrito en la Sección 3.1 ha servido para probar el funcionamiento del MI descrito en la Sección 3.2. En base al conocimiento generado, el MI ha logrado traducir consultas sencillas. No obstante, se ha identificado que para mejorar el desempeño del MI es necesario mejorar la representación semántica mostrada en la Fig. 3 y mejorar el procesamiento semántico que el MI realiza.

5. Conclusiones y trabajos futuros

En la Sección 5.1 se presentan las conclusiones obtenidas a partir del desarrollo de este proyecto y en la Sección 5.2 se presentan los trabajos futuros a realizar a corto plazo.

5.1. Conclusiones

Las ILN son excelentes herramientas para que usuarios inexpertos accedan a información almacenada en bases de datos, ya que no requieren que éstos tengan conocimientos sobre bases de datos, ni necesitan aprender un lenguaje en especial. Su facilidad de uso las hace idóneas para esta labor.

A pesar de que su desarrollo inició hace más de cuatro décadas, aún no han logrado alcanzar el desempeño esperado por los usuarios.

Una pieza clave para lograr un buen desempeño es el conocimiento del dominio, así como el lingüístico, que una ILN debe poseer. Adicionalmente, el modelado de los elementos que componen el conocimiento y el mecanismo de representación de conocimiento juegan un papel importante.

5.2. Trabajos futuros

En primera instancia se trabajará en mejorar la representación semántica mostrada en la Fig. 3 y mejorar el procesamiento semántico que realiza el MI para mejorar su desempeño.

Otro aspecto a abordar a corto plazo es integrar el uso de rebanadores en el proceso de la generación de la consulta MDX. Los rebanadores se deben agregar a la cláusula WHERE. Los rebanadores pueden ser elementos del mismo tipo que los elementos que constituyen la cláusula SELECT. Por tal motivo, integrar el uso de rebanadores implica mejorar el procesamiento de lenguaje natural que el MI realiza, ya que se debe determinar si un elemento identificado se debe agregar a la cláusula SELECT o a la cláusula WHERE.

Posteriormente, se evaluará el desempeño del MI migrando algunas bases de datos utilizadas en la literatura de ILNs a cubos multidimensionales.

Esto debido a que algunas de estas bases de datos están disponibles en la literatura y disponen de un corpus de consultas para realizar la evaluación. No obstante, se analizará la creación de un corpus de consultas para la base de datos multidimensional

AdventureWorksDW2014, ya que posee un número de tablas considerable que implica un reto para cualquier ILN. Las métricas que se utilizarán para evaluar el desempeño serán las métricas *Precision* y *Recall*, comúnmente utilizadas en la literatura para la evaluación de este tipo de interfaces.

Finalmente, a mediano plazo se diseñará un módulo que permita a los usuarios gestionar el conocimiento del MI con el fin de mejorar su funcionamiento y asegurar su utilidad dentro del área de Inteligencia de Negocios.

Agradecimientos. Agradecemos a PRODEP por el apoyo brindado al proyecto titulado *Conocimiento Semántico en una Interfaz de Lenguaje Natural Portable para Acceder a Información de Bases de Datos Multidimensionales en el Área de Negocios Inteligentes*, con el cual este artículo fue posible. UACJ-PTC-373.

Referencias

1. Kuchmann-Beauger, N., Aufaure, M.: A Natural Language Interface for Data Warehouse Question Answering. In: Proceedings of the 16th International Conference on Applications of Natural Language to Information Systems, pp. 201–208 (2012)
2. Saias, J., Quaresma, P., Salgueiro, P., Santos, T.: BINLI: An Ontology-Based Natural Language Interface for Multidimensional Data Analysis. In: Intelligent Information Management, pp. 225–230 (2012)
3. Prat, N., Akoka J., Wattiau, I.: Transforming Multidimensional Models into OWL-DL Ontologies. In: Proceedings - International Conference on Research Challenges in Information Science, pp. 1–12 (2012)
4. Popescu, A., Armanasu, A., Etzioni, O., Ko, D., Yates, A.: Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability. In: Proceedings of the 20th international conference on Computational Linguistics, pp. 141 (2004)
5. Stanford Log-linear Part-Of-Speech Tagger, https://nlp.stanford.edu/software/tagger.shtml (2018)
6. Freeling Features, http://nlp.lsi.upc.edu/freeling/node/4 (2018)
7. SPARQL Query Language for RDF, https://www.w3.org/TR/rdf-sparql-query/ (2018)