

Generación de grafos conceptuales

Sonia Ordoñez-Salinas¹ y Alexander Gelbukh²

¹Universidad Distrital F.J.C and Universidad Nacional, Colombia.
sordonez@udistrital.edu.co

²Instituto Politécnico Nacional, México.
www.Gelbukh.com

Resumen. Los grafos conceptuales se constituyen como formalismo muy especial: además de ser ricos semánticamente, cuentan con algunas características propias de la lógica matemática. Los grafos conceptuales son un formalismo catalogado dentro de las estructuras conceptuales que permiten representar conocimiento por medio de relaciones, conceptos y descripciones a través de símbolos lógicos. En este capítulo, después de una breve introducción a los grafos conceptuales, presentamos los métodos y herramientas utilizados para su construcción manual así como para su generación automática a partir de lenguaje natural.

Palabras Claves: grafos conceptuales, generación automática de grafos conceptuales, semántica, lenguaje natural.

1 Introducción

La mayor parte de la información digital está expresada en forma de lenguaje natural. Una de las grandes necesidades actuales es la de poder gestionar dicha información: almacenarla, consultarla, entenderla y actualizarla. Los medios computacionales se han quedado cortos a la hora de dicha gestión, toda vez que aún los resultados obtenidos no dan los resultados deseados.

Para que los medios computacionales puedan gestionar esta información se ha tenido que recurrir a ciertos artificios estructurales que permiten estandarizar y aplicar algoritmos computacionales. Una de las ventajas de trabajar con una estructura es la estandarización del contenido, lo que significa que a través algoritmos se puede procesar gran cantidad de información, ya que si se quisiera implementar un algoritmo para interpretar el lenguaje natural en su forma cruda, casi que se requeriría una instrucción por cada sentencia específica.

Los grafos conceptuales (como formalismo estructurado y clasificado como estructura conceptual) permiten representar conocimiento a través de aristas entre dos tipos de nodos: conceptos y relaciones, palabras y símbolos propios de la lógica matemática y del lenguaje natural.

Teniendo en cuenta las ventajas de los grafos conceptuales, este capítulo presenta las herramientas, métodos y formalismos utilizados para la obtención de dichos grafos.

El capítulo es estructurado como sigue. En la sección 2 se definen los grafos conceptuales. En la sección 3 se presentan las características principales de los grafos conceptuales. En la sección 4 se da una breve revisión de las principales herramientas que se han desarrollado para la construcción manual de los grafos conceptuales. En la sección 5 se presentan los métodos utilizados para generar los grafos conceptuales automáticamente a partir de los textos en lenguaje natural. Finalmente, la sección 6 concluye el capítulo.

2 Definición de los grafos conceptuales

Durante los años 60 la representación semántica basada en grafos fue popular tanto a nivel teórica como en la lingüística computacional, lo que lleva a la aparición de las redes semánticas [16], redes de correlaciones, grafos de dependencias y los grafos conceptuales con Sowa [45] basados en los grafos existenciales de Pierce [40]. Un grafo conceptual (GC) es una representación gráfica para la lógica basada en redes semánticas y grafos existenciales. Varias versiones de los GC se han implementado en los últimos 30 años y a partir de la investigación de los GC, se han explorado nuevas técnicas de razonamiento, representación de conocimiento y semántica del lenguaje natural.

El estándar “Formato para el modelamiento de esquemas conceptuales” (ISO/IEC 14481) especifica la sintaxis, la semántica y la representación de cadenas de caracteres para el intercambio de grafos conceptuales (*Conceptual graph interchange form, CGIF*) [27] a nivel interno y externo. Las representaciones externas se definen para la comunicación humano máquina y la interna para la comunicación máquina [45].

3 Características de los grafos conceptuales

Los grafos conceptuales para representar texto fueron introducidos por Sowa [47] y los define como grafos bipartitos, donde los nodos pueden representar conceptos y relaciones. Los conceptos tienen un tipo (clase de concepto) y un referente (la instancia de este tipo de concepto). Los nodos relaciones señalan la manera en que los conceptos se relacionan. La relación se establece a través del tipo relacional y la valencia. El tipo relacional indica el rol semántico que realizan los conceptos adyacentes a la relación y la valencia indica el número de estos [45]. Cada relación conceptual tiene uno o más arcos, cada uno de los cuales debe estar enlazado a un concepto. Los conceptos son descritos por palabras y símbolos matemáticos. Un grafo puede relacionarse con otro y un conjunto de grafos pueden constituir un contexto.

Dado que la representación por medio de un grafo conceptual, denota los términos que contribuyen a la semántica de la sentencia y que cada término se escoge de acuerdo a la posición dentro de la sentencia [39], los grafos conceptuales cuentan con una serie de características que hacen que sean muy ricos semánticamente y se utilicen no solo para el intercambio de información sino para la creación de bases de conocimiento y ontologías. Dentro de las características más relevantes de los GC se pueden resaltar las siguientes [30, 46]: a) Los conceptos como las relaciones se pueden catego-

rizan por tipos lo que da una posibilidad de combinaciones e interpretaciones; b) Los conceptos pueden ser especificados por referentes que a través de un cuantificador, un apuntador y un descriptor, determinan las conexiones entre el formalismo del GC y las entidades a que este se refiere; c) Los GC permiten involucrar el contexto a través de un concepto cuyo apuntador es un grafo conceptual; d) Los GC permiten incluir una base de conocimiento cuyo apuntador es un GC consistente de cuatro conceptos: un tipo jerárquico, una relación jerárquica, un catalogo de individuos y un contexto externo.

4 Herramientas para la construcción de los grafos conceptuales

Se ha desarrollado una gran variedad de herramientas que pretenden ayudar en la tarea de la creación de grafos conceptuales. La mayoría se ha constituido como editores de grafos conceptuales con interfaces muy estandarizadas. A continuación se hace una pequeña descripción de algunas de ellas.

PROLOG+CG. Prolog+CG fue desarrollada por A. Kabbaj [21] y nace como una implementación en Java del estándar Prolog, pero con extensiones para la construcción de grafos conceptuales bajo la teoría de Sowa. La herramienta permite utilizar conceptos de programación orientada a objetos, procesamiento de lenguaje natural y lenguaje lógico para proveer a través de los grafos conceptuales un lenguaje de representación de conocimiento que incluye entre otras características redes semánticas, manejo de jerarquías y operaciones de grafos conceptuales. La creación de los grafos se logra a través de la definición de clases.

En las versiones iniciales de Prolog+CG, los grafos conceptuales simples y compuestos se tratan como estructuras básicas y primitivas tales como listas y términos. Posteriormente, Prolog+CG se integra dentro de la plataforma Amine, y en recientes versiones permite incluir los grafos conceptuales dentro de proyectos java. Como resultado de dicha integración, los programas de Prolog+CG son interpretados de acuerdo a una ontología que debe ser cargada primero. Los tipos jerárquicos y las estructuras conceptuales pueden buscarse directamente desde la ontología y cada identificador se puede buscar a partir de un léxico de la ontología. Se introduce la noción de proyecto que pueden compartir los mismos recursos, entre ellos la ontología. A través del lenguaje y los métodos propios de la herramienta se puede crear una ontología que responde a preguntas formuladas por el usuario. La herramienta gráfica se presenta como un editor gráfico para la creación de grafos.

AMINE. Amine es una plataforma de código abierto que permite el desarrollo de diferentes tipos de sistemas inteligentes. Es el resultado de la integración de grafos conceptuales y sistemas inteligentes. Entendida la ontología como aquella que permite describir los conceptos y las relaciones entre los conceptos en un dominio particular. En los GC, una ontología consiste de una jerarquía que contiene tipos que representan grupos de entidades con similares rasgos [1]. Amine está basado en una estructura multinivel que ha progresado en el tiempo. A través de la herramienta se pueden crear

y editar ontologías. A las ontologías definidas como estructuras conceptuales, se le pueden asociar varios léxicos conceptuales, como inglés, francés, español, árabe, etc.

CharGer. Permite crear ontologías definidas como GC en forma gráfica y permite generar su forma lineal. Las instancias son creadas haciendo un enlace al formato de texto y son mostradas en una caja rectangular con el nombre del tipo de concepto. Dentro de las características con las que cuenta esta herramienta se pueden destacar [9]: a) Almacena grafos en diferentes formatos incluido XML; b) Copia y pega los grafos usando un editor interno; c) Los tipos y relaciones jerárquicas pueden ser editadas y almacenadas; d) Soporta los contextos y acceso a bases de datos a través de los actores; e) Soporta algunas operaciones entre grafos como empalme y uniones; f) Permite enlazar a tesauros como Wordnet [29] o glosarios genéricos para conceptos y tipos; g) Permite manejar parámetros de estilo y colores; h) Dentro de las limitaciones que se mencionan en el manual es la falta de mecanismos de validación, así como de copiar y pegar.

GET. Esta herramienta (*Graphs Editor and Toolset*) [52], está definida como una implementación parcial de las estructuras conceptuales de Sowa para sentencias en portugués. GET es un editor de grafos que permite mediante una colección de predicados Prolog implementar operaciones. Permite analizar y generar la notación lineal de los GCs (a partir de un grafo ya definido) como una notación extendida de la presentada por Sowa, especialmente en lo concerniente a tipos y campos en contextos anidados. Como característica especial de esta herramienta, admite incorporar una base de datos que puede hacer referencia a un tipo jerárquico, a un conjunto de grafos y a varios esquemas para un tipo de concepto.

COGITANT. Es una herramienta desarrollada en C++ y nace como extensión de la librería *Conceptual Graphs Integrated Tools* (CoGITO) desarrollada en 1994 [13]. Permite la construcción de bases del conocimiento u ontologías basadas en GC. Provee las clases necesarias para gestionar cada uno de los elementos que hacen parte de los GC, como los mismos grafos, conceptos, relaciones, reglas, restricciones y los métodos para las principales operaciones del modelo de grafos como la proyección. Admite construir los grafos en memoria, los grafos simples o anidados, tipos de conceptos y relaciones. La librería cuenta con los métodos para el acceso a la aplicación desde un aplicativo Java y XML.

COGUI. Es una herramienta gráfica libre, desarrollada en Java para la construcción de ontologías basadas en grafos conceptuales [13]. La construcción de los grafos conceptuales se logra a través de un árbol o de una interfaz gráfica. Permite incluir jerarquías de conceptos, relaciones, reglas y grafos. Cada uno de los elementos que hacen parte del grafo se les puede editar, incluir información adicional, borrar y arrastrar. Para el manejo de la herramienta se crean proyectos y módulos. El editor permite navegar a través de la ontología y editar gráficamente su estructura y contenido. Esta herramienta es heredera de COGITANT con un enfoque a procesos educativos.

CORESE. *Conceptual Resource Search Engine (CORESE)*, es un motor que habilita el procesamiento de instrucciones RDFS, OWL y RDF basado en GC. CORESE está desarrollado en Java y cuenta con un API que permite a los desarrolladores adicionar semántica a las aplicaciones [7]. La principal funcionalidad de la herramienta está dirigida a recuperar recursos Web anotados en *Resource Description Framework, RDF* [51]. RDFs, usando un lenguaje de consulta basado en *Query Language for RDF*, SPARQL y un motor de reglas de inferencia.

El API cuenta con los métodos necesarios para crear y gestionar ontologías, conceptos, propiedades, reglas, anotaciones, instancias, etiquetas, entre otros. Es así, que por ejemplo, A través de la unión de varios grafos, se puede inferir un solo grafo conceptual que permite responder las preguntas formuladas por el usuario. Las transformaciones se logran gracias a las similitudes que existen entre las páginas Web basadas en RDFs y los GCs.

CPE. (*Conceptual Programming Environment*). CP fue originalmente desarrollado como un aplicativo para el desarrollo de sistemas de conocimiento con base en grafos conceptuales y con la posibilidad de manejar información temporal y espacial. A partir de CP surge CPE, con un diseño más modular, flexible y con la posibilidad de interactuar con otras aplicaciones a través lenguajes como el C y el C++. Dentro de las características se pueden resaltar: a) Aplica conocimientos de estructuras de bases de datos a la forma de estructurar y analizar los grafos conceptuales; b) Divide la base de conocimiento en dos capas lógicas: el conocimiento del mundo y el conocimiento del prototipo y c) Permite realizar operaciones básicas de los GCs incluidas la proyección y la de máxima unión [33].

NOTIO. Es un API desarrollado en Java que contienen un conjunto de primitivas para la construcción y manipulación de GCs y provee una plataforma para el desarrollo de herramientas y aplicaciones de GC. Es desarrollada en el 1999, por lo que se presenta como una de las primeras herramientas orientadas a objetos y con la posibilidad de contar con una capa de aplicación que interopera con las demás capas, entre las cuales está la de gestión de los GC [44].

WebKB. Esta herramienta cuenta con una interfaz en línea que permite a través de sentencias especializadas crear o compartir una base de conocimiento. La base de conocimiento fue inicializada con el contenido de la base léxica WordNet [29], sin tener en cuenta la información con respecto a los verbos, adverbios y adjetivos. Se complementaron las categorías de mayor nivel del WordNet pasando a más de 100 tipos de conceptos y 140 tipos de relaciones básicas [28]. Incluye las ontologías como parte de una base de conocimiento, definida como una lista de categorías y declaraciones que dan significado formal a las categorías. Para la herramienta una categoría es una referencia a una clase de objeto (tipo de concepto), relación entre objetos (tipo de relación) u objeto particular (individuo). Cada categoría se asocia con un identificador único que permite distinguir una de otra categoría, y uno o varios nombres. Se maneja un nombre de categoría que puede tener varios significados y que puede estar asociado a varias categorías.

La herramienta integra una serie de lenguajes para su funcionalidad tales como: a) Lenguaje de comandos (*For Structuration, FS*); b) Formato lineal de GC (*Conceptual Graph Linear Format, CGLF*); c) Formato de intercambio de GC (*Conceptual Graph Interchange Format, CGIF*) [27] y d) Lenguaje de descripción de recursos (*Resource Description Format, RDF*).

Dentro de las características del WebKB-2 que se presentan en [26] se pueden destacar: a) Es mucho más amplio y consistente que el WordNet; b) Es expandible en cualquier momento por cualquier usuario, c) Por estar en línea, permite cooperación asíncrona entre los usuarios ya que los usuarios pueden rehusar, complementar o corregir el conocimiento de otros usuarios pero no tienen que estar de acuerdo entre sí; d) La base de conocimiento es única y al ser compartida en línea, permite al máximo la interconexión de conocimiento, recuperación y descubrimiento de inconsistencia; e) Los lenguajes de representación de conocimiento utilizados orientan al usuario a expresar y leer conocimiento; f) Limita la forma de expresar el conocimiento estableciendo un lenguaje para tal fin; g) Permite la construcción compartida de páginas amarillas, como listas de catálogos; h) Puede recuperar documentos que incluye palabras claves y permite extraer conocimiento, es decir recuperar e interrelacionar información precisa.

Ontoseek. (*Content-Based Access to the Web*). Ontoseek utiliza el WordNet y los grafos conceptuales existenciales para crear páginas amarillas o catálogo de productos. En el trabajo [14] demuestran que con el uso de la estructura propuesta, lenguaje de expresiones limitadas y la explotación de ontologías lingüísticas como el WordNet, se incrementa el índice de precisión y de recuperación. Permite realizar búsquedas clásicas, por ejemplo, por nombre en cambio de categorías.

Ontolingua. Ontolingua es una herramienta que soporta más de 150 usuarios activos en línea y en ambiente distribuido. Permite buscar, crear, revisar, modificar y usar ontologías [12]. Admite que aplicaciones se conecten remotamente al servidor a través de una interfaz especial. La página de la herramienta presente una fecha del año 1995 y algunos de los enlaces están deshabilitados, por lo que se expondrá muy brevemente en qué consiste. De acuerdo a la documentación, proporciona un conjunto de herramientas y librerías modulares que permiten la administración de las ontologías. Soporta la inclusión de grafos cíclicos y habilita a los usuarios para extender las ontologías en múltiples formas. Los instrumentos en Ontolingua se orientan hacia la creación, edición y actualización de ontologías.

CARE. Son dos herramientas, NETCARE y CARE [34] siendo la primera predecesora de la segunda. La diferencia entre las dos radica en que la primera está desarrollada en Prolog y la segunda es desarrollada como un applet de Java. Sin embargo la filosofía de las dos es la misma, es decir que son herramientas que asisten a la construcción gráfica de grafos conceptuales. Permiten modelar bases de datos por medio de GCs, e involucran algunas operaciones del lenguaje de consulta estructurado (*Structured Query Language, SQL*).

Grafitos. Es un editor didáctico para grafos conceptuales que hace parte de un proyecto de grado en el programa Ingeniería de Sistemas y Computación, de la Universidad Tecnológica de Pereira [35]. El objeto de Grafitos es facilitar el aprendizaje de la naturaleza de los grafos conceptuales. Cuenta con una interfaz para la composición y la edición gráfica de los mismos, así como la transformación automática a su representación en formato lineal y viceversa. La herramienta permite: a) Definir y editar de una base de conocimiento (ontología de conceptos y relaciones); b) Representar visualmente la base de conocimiento; c) Componer grafos conceptuales; d) Transformar los grafos conceptuales de la forma gráfica a la forma lineal y viceversa y e) trabajar de modo libre, es decir con restricciones semánticas definidas por el propio usuario.

SOMED. Esta herramienta [4] se enmarca dentro del dominio de la medicina y actualmente se encuentra en su versión 3. Permite esquemas de códigos que incluyen modificadores y referencias de tiempo. Permite manejar grafos conceptuales de forma consistente ya que permite relacionar los códigos con herramientas para modelos relacionales de bases de datos y de cálculo de predicado de primer orden.

5 Transformación automática de texto a grafos conceptuales

No son muchos los trabajos cuyo objeto es la construcción de los grafos conceptuales sin embargo de los encontrados siguen claramente dos tendencias: Los que utilizan un procedimiento determinístico en el sentido que no requieren entrenamiento de entrada y que siguen paso a paso ciertas órdenes y reglas y los que siguen un procedimiento estadístico que por medio de aprendizaje de máquina intentan detectar una o varias partes del grafo conceptual. En la mayoría de los trabajos se utiliza una gramática para obtener una notación estructural inicial y fácil de procesar, se ayudan de léxicos como el WordNet [29] u ontologías y generalmente se incluyen un conjunto de reglas que permiten seguir los pasos secuenciales o realizar el aprendizaje automático.

Métodos determinísticos. Dentro de esta clasificación se han incluido aquellos trabajos que muestran cómo construir un grafo conceptual desde el punto de vista metodológico, estructural o procedimental. En este primer trabajo [17], se muestra cómo construir un grafo a partir de una frase desde el punto de vista metodológico. Mientras que en [31], se presenta un caso de la creación de estos desde el punto de vista de modelamiento ya que se crean los grafos conceptuales similar a como se crean las clases utilizando el lenguaje unificado de modelamiento (UML). En [54] las diagramas UML, que es un formalismo muy cercano en su espíritu a los grafos conceptuales, se obtienen automáticamente a partir del lenguaje natural controlado.

En los trabajos que se mencionan a continuación, con algunas variaciones se construye el grafo conceptual siguiendo un procedimiento lineal. En [8] se presenta un sistema que convierte un texto libre en español a grafos conceptuales. El texto libre pasa por un analizador morfológico [41] y un analizador sintáctico. A partir de los árboles se crean los grafos. Un paso crucial en la construcción de los grafos conceptuales, a diferencia de los árboles sintácticos, es la resolución de anáfora [42]. En [2] se trabaja con textos en francés y los grafos conceptuales se construyen a partir de dos estructu-

ras iniciales. La primera modela la representación semántica que permite a través de sistemas primitivos expresar patrones cognitivos, mientras que la segunda es un diagrama temporal que representa las restricciones temporales y que son exploradas a través de los tiempos de los verbos. Uniendo estas estructuras obtienen el modelo de grafos conceptuales a un nivel básico.

Sobre el dominio específico de la medicina se encuentran trabajos [37, 36], que buscan la estructuración de reportes médicos y los GC se construyen a partir de anotaciones sobre el texto que se consiguen con base en reglas de proximidad, información semántica y sintáctica. En [38], se construye un sistema llamado Metaxa que se enmarca en el dominio de las anotaciones sobre radiologías y se representa el contenido de los reportes en grafos conceptuales.

El sistema presentado en [10], a pesar de estar en desarrollo, puede ser mencionado ya que espera lograr la representación de la semántica para reportes de diagnóstico con base en la nomenclatura multiaxial y la indexación de sustantivos, para luego transformar dichos índices en grafos conceptuales. En [6] se presentan algunos aspectos de cómo se logra transcribir registros de asistencia médica a una estructura de grafos conceptuales, con el fin de realizar mejor consultas y recuperación. El trabajo se centra en el modelamiento de una base de conocimiento centrada en la estructura de grafos conceptuales, muy similar a como se modela bajo un paradigma de entidad/relación.

Métodos estadísticos. En [17, 18, 19] a través del VerbNet [23] y/o el WordNet se identifican los roles semánticos y con base en estos y un conjunto de reglas sintácticas y/o semánticas se construyen los grafos conceptuales. Vale la pena resaltar el [15], un prototipo de un procesador semántico de sentencias italianas que usa un léxico especial manualmente adquirido y que extrae parcialmente conocimiento de los textos a través de plantillas o por la generación de un conjunto de patrones lingüísticos.

El problema planteado en [3] es el de construir automáticamente a partir de texto una base de conocimiento léxico expresada en grafos conceptuales. Para lo cual, etiquetan las palabras, utilizan un analizador gramatical para construir los árboles gramaticales y a partir de una serie de reglas transforman dichos árboles en grafos conceptuales. Una vez transformados se optimizan los grafos al desambiguarlos estructural y semánticamente. Para la construcción de las reglas tanto del analizador gramatical como de las reglas que permiten pasar los árboles gramaticales en grafos conceptuales se utilizan diferentes técnicas incluidos métodos heurísticos.

Como ya se ha notado se utilizan diferentes gramáticas para conseguir una primera aproximación a los grafos. Dentro de los primeros trabajos que utilizan un analizador gramatical para posteriormente obtener los GC, está el de [48]. Utilizando las gramáticas de enlace [25], se pueden citar los siguientes dos trabajos.

Primero, en [53] se construyen los GC a partir de dichas gramáticas, el WordNet y las máquinas de aprendizaje como un problema de clasificación que puede ser entrenado para diferentes dominios y un conjunto pequeño de reglas. En el trabajo [43] se aprovecha la similitud que existe entre las gramáticas de enlace y los grafos conceptuales, para formular un problema de clasificar las estructuras conceptuales. Una implementación del sistema fue desarrollada para el laboratorio de investigación de la International Business Machines IBM en China.

Segundo, [22] también utiliza estas gramáticas para producir las relaciones sintácticas entre las palabras de una sentencia e incorpora las condiciones necesarias para proporcionar las necesidades especiales del dominio. Para generar el grafo conceptual, la estructura generada la cruzan con las raíces. El generador de grafos conceptuales consistió de un algoritmo que va identificando los diferentes elementos de la sentencia; El determinante, el adjetivo, el sustantivo, la preposición, el verbo, el sustantivo preposicional, las relaciones, etc.

En [24] se trata de desarrollar un sistema de inducción gramatical que pueda asignar sentencias descriptivas a modelos ontológicos representados por grafos conceptuales. Para esto el sistema aprende las reglas de asociación entre las sentencias y representaciones ontológicas a través de un enfoque de clasificación. Los elementos del modelo son agrupados bajo clases: conceptos, relaciones y contenedores o módulos estructurales. Los conceptos y las relaciones se dividieron adicionalmente de acuerdo a varios aspectos. El grafo conceptual se obtiene a partir de una estructura de dependencia.

El propósito del trabajo presentado en [11] es el de explorar la estructura de la frase para generar hipótesis de conceptos, calificar estos de acuerdo a la probabilidad y seleccionar aquellas más creíbles a través agrupamientos (método no supervisado). La adquisición del conocimiento inicial se logra a través de un analizador basado en la gramática de árbol adjunto y la extracción de dependencias [50] se logra con un analizador de dependencias.

Por último, se puede resaltar las investigaciones relacionadas con las operaciones sobre los grafos conceptuales, como [32], donde se definen algunas medidas para la semejanza entre los grafos conceptuales, así como los trabajos que como [5] que tienen como objetivo la construcción automática de los léxicos semánticos necesarios para la construcción correcta de los grafos conceptuales.

6 Conclusiones y trabajo futuro

Si bien los Grafos Conceptuales han mostrado tener bondades en la inclusión de la semántica, por lo que no solo se permite la representación del lenguaje natural, sino la creación de ontologías y bases de conocimiento. Aún los métodos para gestionar estas estructuras esta por explorar, pues aún cuando se han presentado tanto investigaciones como herramientas se puede concluir: a) No se encuentran colecciones de grafos conceptuales que permitan experimentar con ellos; b) Las herramientas desarrolladas se presentan a modo de editores gráficos y no permiten generar automáticamente el grafo conceptual a partir de un texto expresado en lenguaje natural. c) Los APIs desarrollados permiten crear las estructuras de los grafos conceptuales a manera generalmente de clases; d) Las investigaciones revisadas describen los métodos para generar automáticamente los grafos conceptuales, pero los experimentos no se pueden reconstruir ya que no aparecen los grafos creados ni las colecciones utilizadas.

Agradecimientos. Este trabajo es financiado parcialmente por la *Universidad Distrital F.J.C.*, Bogotá, Colombia, y parcialmente por el gobierno de México (CONACYT 50206-H, SIP-IPN 20100773, SNI).

Referencias

1. Abdulrub S., Polovina S., Hill, R., Implementing Interoperability through an Ontology Importer for Amine. Conceptual Structures Tools and the Web. Third Conceptual Structures Tool Interoperability Workshop, 2008.
2. Amghar, T., Battistelli, D., Chamois, T. Reasoning on aspectual-temporal information in French within conceptual graphs. Tools with Artificial Intelligence, 2002. 14th IEEE International Conference ICTAI 2002, pp. 315–322, 2002.
3. Barrière, C. From a Children's First Dictionary to a Lexical Knowledge Base of Conceptual Graphs. Tesis Doctoral. Ecole Polytechnique de Montreal. 1997.
4. Campbell, K. E., Musen M. A. Representation of clinical data using SNOMED III and conceptual graphs. Proceedings of the Annual Symposium on Computer Application in Medical Care. pp. 354–358. 1992.
5. Castro-Sánchez N.A., Sidorov G. Analysis of Definitions of Verbs in an Explanatory Dictionary for Automatic Extraction of Actants based on Detection of Patterns. Lecture Notes in Computer Science 6177, pp 233–239, 2010.
6. Chang C.S., Chen A.L.P., Supporting Conceptual and Neighborhood Queries on WWW, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Application and reviews, Vol. 28, No. 2, pp. 300–308. 1998.
7. Corby O., Dieng-Kuntz R., Faron-Zucker C. Querying the Semantic Web with the CORESE search engine. Lopez de Mantaras R., L. Saitta L., (ed.). Proc. of the 16th European Conference on Artificial Intelligence (ECAI'2004), sub-conference PAIS'2004, Valencia. pp. 705–709, 2004.
8. Cruz, M. H. Generador de los grafos conceptuales a partir del texto en español. Tesis de Maestría. Instituto Politécnico Nacional, México, 2007.
9. Delugach H., A Conceptual Graph Editor, CharGer Manual v3.5b1. User's Guide, 2005.
10. Denecke K., I., Kohlhof, J. B. Use of Multiaxial Indexing for Information Extraction From Medical Texts Integrating Biomedical Information: From E-cell to E-patient. Proceedings of the European Federation for Medical Informatics Special Topic Conference 2006. IOS Press. April, 2006.
11. Farkas, J., Improving the classification accuracy of automatic text processing systems using context vectors and back-propagation algorithms. Electrical and Computer Engineering, 1996. Canadian Conference. Vol.2, pp. 696–699, 1996
12. Fikes R., Farquhar A., Rice J., Tools for Assembling Modular Ontologies in Ontolingua. Knowledge Systems, AI Laboratory. 1997.
13. Genest, D., Salvat, E. A platform allowing typed nested graphs: How CoGITo became CoGITaNT. In: Mugnier, M. *et al.* (eds.) Conceptual Structures: Theory, Tools and Applications. Lecture Notes in Computer Science, Springer, pp. 154–161. 1998.
14. Guarino, N.; Masolo, C., Vetere, G. OntoSeek: Content-Based Access to the Web. IEEE Intelligent Systems, IEEE Educational Activities Department, 1999.
15. Harabagiu S., Maiorano S. Acquisition of linguistic patterns for knowledge-based information extraction. Proceedings of LREC 2000, Athens. 2000.
16. Helbig H. Knowledge Representation and the Semantics of Natural Language. Springer. 2006.
17. Hensman S. Construction of Conceptual Graph representation of texts. Department of Computer Science, University College Dublin. Belfield, Dublin 4. Proceedings of Student Research Workshop at HLT-NAACL. 2004.

18. Hensman, S., Dunnion, J. Automatically building conceptual graphs using VerbNet and WordNet. 2004 international Symposium on information and Communication Technologies. ACM International Conference Proceeding Series, vol. 90. Trinity College Dublin. Las Vegas. pp. 115–120. 2004.
19. Hensman, S., Dunnion, J., Constructing conceptual graphs using linguistic resources. 4th WSEAS international Conference on Telecommunications and informatics. Husak M. and Mastorakis N., (Ed). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin. pp. 1–6. 2005.
20. ISO/IEC 14481: Information Tecnology (IT). Conceptual Schema Modeling Facilities CSMF. 1998.
21. Kabbaj, A., Janta-Polczynski, M., From PROLOG++ to PROLOG+CG: A CG Object-Oriented Logic Programming Language in Conceptual Structures: Logical, Linguistic, and Computational Issues, Lecture Notes in Computer Science. Editor Bernhard G. and Guy M., Springer, Vol. 1867. pp. 540–554. 2000.
22. Kamaruddin, S.S.; Bakar, A.A.; Hamdan, A.R.; Nor, F.M., Conceptual graph formalism for financial text representation, Information Technology, 2008. ITSIM 2008. International Symposium , Vol.3. pp.1–6. 2008.
23. Kipper K., Korhonen A., Ryant N., Palmer M.. Extending VerbNet with Novel Verb Classes. Fifth International Conference on Language Resources and Evaluation, LREC 2006. Genoa, Italy. June, 2006.
24. Kovacs, L., Baksa-Varga, E., Dependency-based mapping between symbolic language and Extended Conceptual Graph, Intelligent Systems and Informatics, 2008. SISY 2008. 6th International Symposium, Sept. 2008, pp. 1–6.
25. Lafferty J., Sleator D., Temperley D., Grammatical Trigrams: A Probabilistic Model of Link Grammar. Proceedings of the AAAI Conference on Probabilistic Approaches to Natural Language. October, 1992
26. Martin P. A., Eklund P. Large-scale cooperatively-built heterogeneous KBs. ICCS, 9th International Conference on Conceptual Structures, Stanford University, California, LNAI 2120 Springer Verlag USA. pp. 231–244. 2001.
27. Martin, P. Knowledge Representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English, Conceptual Structures: Integration and Interfaces. Lecture Notes in Computer Science. Springer, Vol. 2393. pp. 77-91. 2002
28. Martin, P.A. and Eklund, P. WebKB. Proceedings of the 7th International Conference on ,Conceptual Structures. 1999.
29. Miller G., A., WordNet: A Lexical Database for English. Communications of the ACM, Vol. 38., No. 11. pp. 39-41. 1995
30. Mineau G. W, Moulin B., Conceptual Grpahs for Knowledge Representation, First International Conference on Conceptual Structures, ICCS'93. Lectures Notes in Artificial Intelligence. Sowa J.F. (Ed.), Springer, 1993.
31. Mineau, G. W., Stumme, G, Wille, R. Conceptual Structures Represented by Conceptual Graphs and Formal Concept Analysis, International Conference on Conceptual Structures, 1999.
32. Montes y Gómez M. Minería de Texto Empleando la Semejanza entre Estructuras semánticas. Tesis Doctoral. Instituto Politécnico Nacional. Laboratorio de Lenguaje Natural y Procesamiento de Texto, México. 2002.
33. Pfeiffer D, Chavez N., Heather D., Pfeiffer J. CPE Design Considering Interoperability, New Mexico State University, 2004.
34. Polovina, S. CARE: A Practical Conceptual Graphs Software Tool. 2007.

35. Quijano B. H., Espinosa M. A. Grafitos: un editor didáctico para grafos conceptuales. programa Ingeniería de Sistemas y Computación, de la Universidad Tecnológica de Pereira. 2004.
36. Rassinoux, A. M., Baud, R. H., Lovis, C., Wagner, J. C., Scherrer, J. R. Tuning Up Conceptual Graph Representation for Multilingual Natural Language Processing in Medicine. 6th International Conference on Conceptual Structures, ICCS'98. Conceptual Structures: Theory, Tools, and Applications. Lecture Notes in Computer Science. Springer. Vol. 1453, 1998.
37. Rassinoux, A. M., Baud, R. H., Scherrer, J. R. A Multilingual Analyser of Medical Texts. Second International Conference ICCS'94. Conceptual Structures: Theory, Tools, and Applications. Lecture Notes in Computer Science, Springer, Vol. 835. pp. 84-96. 1994.
38. Schröder, M. Knowledge based analysis of radiology reports using conceptual graphs. Conceptual Structures: Theory and Implementation. Lecture Notes in Computer Science Springer. Vol. 754. pp. 293-302. 1993.
39. Shehata, S., Karray, F., Kamel, M. Enhancing Text Retrieval Performance using Conceptual Ontological Graph. ICDM Workshops 2006. Sixth IEEE International Conference. 2006.
40. Shin S.-J. The Logical Status of Diagrams Cambridge University Press, 1994.
41. Sidorov G. Etiquetador Morfológico y Desambiguador Manual: Dos aplicaciones del analizador morfológico automático para el español. En VI encuentro internacional de computación ENC-2005, México, pp. 147-149, 2005.
42. Sidorov G., Olivas Zazueta O. Resolución de anáfora pronominal para el español usando el método de conocimiento limitado. In: "Avances en la Ciencia de la computación", 7º congreso internacional ENC-2006, México, pp. 276-281, 2006.
43. Sleator, D. D., Temperley, D. Parsing English with a link grammar. Third International Workshop on Parsing Technologies. 1993.
44. Southey, F., Linders, J. G., Notio. A Java API for developing CG tools in Conceptual Structures: Standards and Practices, Springer. pp. 262-271. 1999.
45. Sowa J., F Conceptual Graph Standard. Committee on Information Interchange and Interpretation. 2000.
46. Sowa J., F, Common Logic. A Framework for a Family of Logic-Based Languages. 2008.
47. Sowa J., F. Conceptual Estructures, Information Processing in Mind and Machine, The Systems Programing Series Addison-Wesley. 1984.
48. Sowa J.F., Way, E. C. Implementing a semantic interpreter using conceptual graphs. IBM J. Res. Dev. 30. pp. 57-69. 1986.
49. Sowa, J. F., Conceptual Graphs. Handbook of Knowledge Representation., Foundations of Artificial Intelligence., Vol. 3, Harmelen F., V., Lifschitz V., Porter B., (Eds.), Elsevier, pp 213-237. 2008.
50. Tesnière, A. L. Elements de syntaxe structurale. Libraire C. Klincksieck, 1959.
51. W3. RDF/XML Syntax Specification. W3C Recommendation., Dave Beckett, (ed). 2004.
52. Wermelinger, M. GET: An Implementation of Conceptual Graphs. Centro de Inteligencia Artificial / UNINOVA Portugal. 1991.
53. Williams, R. A., Computational Effective Document Semantic Representation. Digital EcoSystems and Technologies Conference, 2007. DEST'07. Inaugural IEEE-IES. pp. 410-415. 2007.
54. Zapata Jaramillo C. M., Gelbukh A., Arango Isaza F. UN-Lencep: Obtención Automática de Diagramas UML a partir de un Lenguaje Controlado. Avances en la Ciencia de la Computación. 7º congreso internacional ENC'06, 2006.