



Cuando las **MÁQUINAS** entiendan lo que **leen**

Alma Delia Cuevas

La comprensión de textos por las computadoras es aún limitada. Sin embargo, se prevé que aprendan a “leer” documentos, “fusionando”, “juntando pedazos del conocimiento”, uniendo información procedente de textos, escritos y documentos. Actualmente se trabaja con programas automáticos para lograr que la computadora se convierta en un “sabelotodo”.

Imagine que una computadora pudiera *entender* (en el sentido de “usar para responder correctamente preguntas no triviales”) un área extensa del conocimiento humano. Por ejemplo, leer (y entender) un documento sobre las amapolas, o una reseña de *Cien años de soledad*. ¿Tendría esto alguna ventaja?

Claro que sí, pero... ¿y qué utilidad tendría guardar conocimiento *entendible por la máquina* (en el sentido ya dicho) en una estructura de datos —una *ontología*, para ser precisos— (por ejemplo, información de un área extensa, digamos, física o finanzas)?

Sería de gran utilidad: permitiría que las personas (y *también las máquinas*) hicieran deducciones, extrapolaciones, sacaran conclusiones y formularan hipótesis a partir de esta gran cantidad de conocimientos. Tendríamos un “aliado” que no sólo es un bibliotecario que nos guarda información para que nosotros la leamos y entendamos, sino que ahora es un “sabelotodo” (o un “sabe física”) que nos puede brindar respuestas a preguntas no triviales sobre un cierto tema.

La comprensión de textos por computadoras es ahora limitado. Hoy en día, una computadora puede subrayar con una rayita roja una palabra escrita con mala ortografía, y con una rayita verde una oración que sea incorrecta gramaticalmente. Si la computadora pudiera *entender* lo que estamos escribiendo (o leyendo), nos podría subrayar con morado cuando una frase no fuera cierta, es decir, cuando no concordara con los datos externos (“Napoleón nació en Salina Cruz, Oaxaca”, o “El autor

de *La guerra y la paz* es Dostoievski”, o “Veracruz es la capital de Xalapa”). Si un documento dice que Veracruz es la capital de Xalapa, así será representado en la ontología correspondiente. Que la máquina detecte este error depende de si ya sabe que Xalapa es una ciudad.

Cómo representar el conocimiento

La información, en su naturaleza numérica, se ha representado en las bases de datos desde hace décadas; la información no numérica, en cambio, se ha representado hasta hace algún tiempo a través de las ontologías.

Aunque actualmente la forma más común de representar y guardar la información en las computadoras es a través de las *bases de datos relacionales* (véase artículo “La información es poder... sobre todo si está en una base de datos”, en este mismo número de *Ciencia*), es mejor, para hacer deducciones sobre datos y conceptos no numéricos, representar la información como una red de conceptos, cuyos nodos son los objetos (“sustantivos” como *Porfirio Díaz* o *armadillo*), y los *enlaces*, lados o aristas de la red, son relaciones (“procesos”, verbos o acciones, como *fundó* o *brincaba*). A estas estructuras de datos (redes semánticas) se les llama *ontologías*.

Las ontologías son una manera de guardar información usando una estructura de datos. Consisten en un conjunto de nodos, conceptos, ideas, elementos (representados por círculos) que se encuentran relacionados entre sí (representados por líneas). Gráficamente, se representan con un grafo con aristas y vértices, como se muestra en la Figura 1.

Se han inventado varios lenguajes para representar estas ontologías. Los más conocidos son: OWL, RDF, DAML+OIL, etcétera. Sin embargo, hasta ahora dichos lenguajes no cubren todas las necesidades de representación de información, sobre todo cuando se trata de especificar los detalles de un texto. Por ejemplo, la representación de clasificaciones específicas de un objeto (partición), es decir, clasificar a un individuo de acuerdo con su edad. (Se llama “partición” porque una misma persona no puede pertenecer a dos clasificaciones diferentes, ser niño y joven a la vez.)

En el Centro de Investigación en Computación (CIC) del Instituto Politécnico Nacional, en México,

hay un grupo que ha inventado su propio lenguaje de representación de ontologías: la notación OM. Esta notación sí considera la clasificación tipo partición. Otra ventaja de la notación OM se da en las relaciones o enlaces que permite representar, ya que pueden ser también conceptos o conjuntos (conocidas como *clases* en OWL). Esta funcionalidad le da más significado al concepto (idea) que se representa en la ontología, y permite representar (contener) la información de un texto común, pese a todos los detalles descritos en él. Otra ventaja importante es que la notación OM representa el texto para que lo entienda la máquina, a diferencia de las notaciones OWL, RDF, que representan el texto para entendimiento de los seres humanos, y eso le resta detalles en su representación.

¿Quién construye estas ontologías? Actualmente, se construyen de manera manual. Las hacen las personas, especialistas en el área en que la información de la ontología refleja. Por tanto, no existe un estándar de

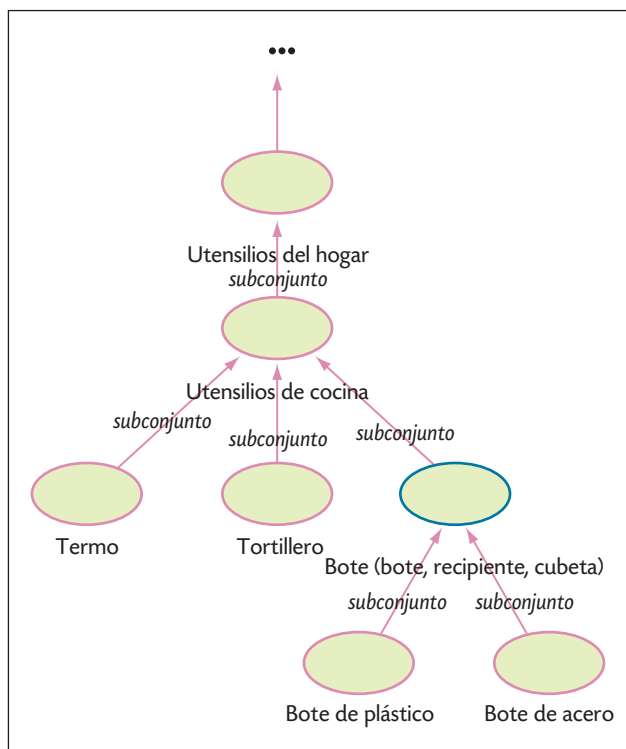


Figura 1. Ontología que indica los nodos (elementos, conceptos, representados por círculos); los títulos en la parte inferior del nodo representan el nombre del concepto, y las palabras entre paréntesis son los sinónimos del concepto. Las flechas son los enlaces entre los nodos; cada enlace tiene un título que representa el nombre de la relación.

construcción: los estilos son diferentes. Si se quisiera diseñar una ontología más cercana al uso cotidiano de la información, entonces las herramientas que apoyarían al proceso de diseño serían diccionarios comunes, de sinónimos y demás clasificadores de temas que son de uso común.

**Un ejemplo:
el proyecto cyc**

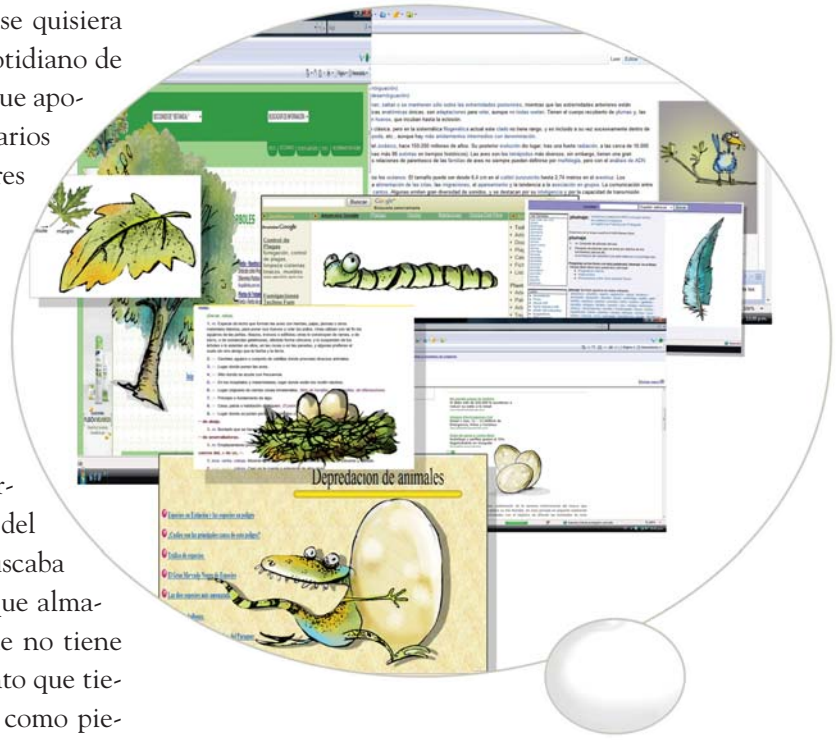
Por esa razón, distintos grupos de científicos se han dedicado a trabajar en el área de las ontologías. Ha habido esfuerzos titánicos, como el del Proyecto CYC (del inglés *enCYClopedia*, Cycorp, 2002), que buscaba hacer “una ontología del sentido común” que almacenara el conocimiento de una persona que no tiene especialización. Por ejemplo, el conocimiento que tiene un niño de nueve años; es decir, temas como piedras, calles, lluvia, automóviles, padres, gatos, etcétera.

¿Para qué? Bueno, así las computadoras sabrían qué cosas son “razonables” en el mundo real. Con esto, la computadora podría leer documentos por sí sola, es decir, *entenderlos* (en el sentido indicado al principio de este artículo).

Durante diez años, los colaboradores de CYC trabajaron incansablemente para buscar este elusivo fin: la representación del conocimiento común. Cuando el proyecto concluyó, sus resultados estaban lejos de lo esperado. Nosotros creemos que las metas de CYC son alcanzables; la dificultad estriba en el proceso (hacer manualmente “una sola ontología” extensa).

Más práctico: construir ontologías pequeñas

Entretanto, otros desarrolladores menos ambiciosos empezaron a hacer ontologías más pequeñas, más modestas y prácticas: sobre vinos, productos lácteos, ventas por catálogo, tipos de impresoras para computadora, etcétera. Estos repositorios de conocimiento eran “superficiales”, con la información contenida principalmente en el nombre y la descripción (en español) de los artículos o conceptos representados, porque estaban destinados principalmente a que los



En el Centro de Investigación en Computación (CIC) del Instituto Politécnico Nacional, en México, hay un grupo que ha inventado su propio lenguaje de representación de ontologías: la notación OM

usaran personas, no máquinas. Con ellas, las computadoras no pueden hacer deducciones sofisticadas. Es como si las computadoras pudieran guardar números, grandes cantidades, pero no pudieran sumarlos o decirme cuánto se vendió ayer en un almacén.

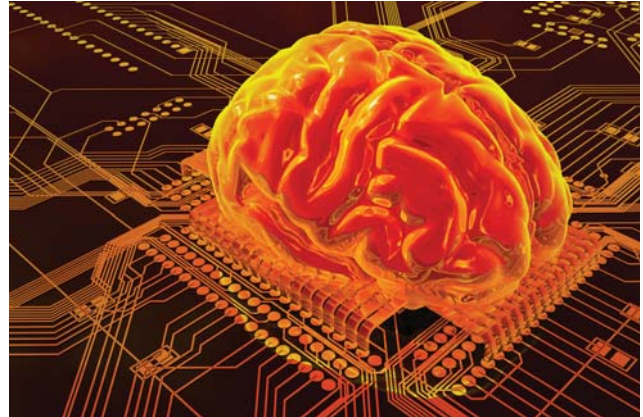
El problema es unir las

Una vez que se diseñan las ontologías pequeñas es preciso que se vayan fusionando entre sí para ir construyendo un conocimiento cada vez mayor, agrupando los temas comunes (o conocimiento común), aunque también se pueden fusionar temas especializados. Por ejemplo, juntar una buena cantidad de ontologías sobre Benito Juárez en una única permitiría (a nosotros y a las computadoras) “conocer más” sobre el Benemérito de las Américas. Antes de fusionar las ontologías se realiza una *alineación* (comparación) entre ellas, donde se identifica el significado de cada elemento de la ontología y su *similitud* (sinonimia) entre los elementos de otras ontologías a fusionar.

Dado que durante el proceso de alineación y fusión surgen problemas con palabras ambiguas (por ejemplo en la decisión de identificar o no el concepto correcto a aplicar), se requiere la intervención del usuario. Hasta recientemente, el usuario, con su sentido común, era quien resolvía la ambigüedad.

Si tenemos dos ontologías sobre el mismo tema (vinos, por ejemplo) pero hechas por distintas personas, quizá con distintos puntos de vista, ¿podríamos juntar o fusionar el conocimiento guardado en ellas, para así “saber más” de un tema dado? La respuesta es sí. De esta manera, una computadora podría “amasar” conocimiento, es decir, concentrar un número considerable de información proveniente de ontologías pequeñas. Se tendría en una computadora, o en un grupo de ellas, el conocimiento de todas las cosas naturales, inventadas, abstractas, concretas, etcétera.

Pero lo más importante es lograr que este proceso de fusión se lleve a cabo de manera automática, es decir, que el programa que fusiona ontologías considere la solución a todos los problemas que puedan surgir durante el proceso; así sería más independiente del usuario. Lo ideal sería hacer que la fusión fuera automática, independiente del usuario.



Hasta hace un tiempo, había dos enfoques para adquirir conocimiento por este método:

- a) Hacer manualmente una ontología bajo uno de los lenguajes de diseño de ontologías, como *Ontology Web Language*, *OWL* o *Resource Description Framework*, *RDF*, por citar algunos).
- b) Usar un *editor de ontologías* que las comparaba y sugería qué conceptos de una podrían corresponder con cuáles conceptos de otra, y permitían que el usuario hiciera la unión o fusión final (como en Dou y colaboradores, 2002; Fridman y Musen, 2000; Stumme y Maedche, 2002, que son proyectos de universidades importantes como Stanford, Yale, Karlsruhe, Alemania, Southampton, Inglaterra y la Universidad del Sureste de California, respectivamente), es decir, estos trabajos realizan la fusión de manera *semiautomática*.

Otro enfoque más audaz es que la fusión la haga un programa de cómputo. La máquina “lee” o digiere las ontologías por sí sola, y va formando otra ontología con el conocimiento de ambas, teniendo cuidado con los sinónimos, homónimos, diferentes clasificaciones, inconsistencias en la información e información incompleta. El proyecto *HCONE-Merge* (Kotis, Vouros y Stergiou, 2005), de la Universidad Egea en Karlovassi, Grecia, es uno de estos ejemplos, ya que se basa en un algoritmo que realiza la fusión de manera automática. Otro trabajo se cita en el proyecto *OM* (Cuevas, 2006) en el que usamos un fusionador automático, el primero de su tipo en el mundo.

Juntar “pedazos del conocimiento” automáticamente

Tuve la suerte de desarrollar un fusionador automático de ontologías (Cuevas, 2006) como parte de un grupo de investigación del CIC. El fusionador, llamado OM, es sencillo, pero su mecanismo interno para fusionar no es fácil. Necesita dos ontologías (fuente) almacenadas en la Notación OM; el programa las carga en memoria y después las fusiona, generando una ontología resultante que contiene la información de las ontologías fuente, sin inconsistencias ni contradicciones, y sin relaciones redundantes.

Antes de ser fusionadas, las ontologías se comparan (proceso llamado alineación en otros trabajos de fusión), y se deben resolver de manera automática los problemas que se puedan presentar. Algunos de estos problemas son:

1. *Identificar cada concepto, nodo o elemento de una ontología a otra.* Es decir, puede haber nodos de una ontología que tengan diferente nombre pero mismo significado, llamados *sinónimos* (caballo y corcel, por ejemplo); el fusionador OM los identifica y los fusiona, enriqueciendo dicho significado.
2. *Identificar nodos polisémicos:* si en ambas ontologías hay un nodo con el mismo nombre pero diferente significado, por ejemplo “manzana” como fruta y “manzana” como ubicación geográfica, el fusionador los identifica como conceptos diferentes y los fusiona considerando la semántica de cada uno de estos nodos.



3. *Identificar nodos con información general y específica.* Por ejemplo, en una ontología está representado que “Agustín Lara nació en Veracruz”, y en otra que “Agustín Lara nació en Tlacotalpan”. El fusionador identifica que ambas cadenas son diferentes y “descubre” que Tlacotalpan es una ciudad que pertenece a Veracruz, y por tanto los fusiona, considerando esta nueva información.
4. *Reconocer diferentes puntos de vista.* Para una ontología, México es una república federal. Para otra, México es un mercado subdesarrollado. ¿Es posible que México sea ambas cosas simultáneamente? Sí, lo es, y OM le da a México dos “padres” o ancestros: “república federal” y “mercado subdesarrollado”.
5. *Identificar varias maneras de cortar el pastel.* Para una ontología, las personas se dividen en hombres y mujeres; para otra, en niños, jóvenes y adultos. Ambos puntos de vista (ambas *particiones*) aparecen en la ontología que OM produce.

Así, OM va identificando cada uno de los nodos de las ontologías fuente y, además de las situaciones arriba descritas, también reconoce otras que no explicamos a detalle por motivos de espacio. Estos detalles le dan robustez e independencia al proceso de fusión, lo cual diferencia el proyecto OM de los *alineadores* (que requieren un ser humano como componente) o de los que construyen ontologías manualmente.

Nótese que OM no es un “guardián de la verdad” o *verificador*: no juzga si una ontología expresa conceptos que concuerdan con la realidad o son mentiras. Si A dice que la Tierra es plana, y B también, la fusión de OM producirá una ontología donde se registra que la Tierra es plana, por ser consistente con A y con B.

La computadora “aprende” juntando ontologías

En este apartado se plantean tres ejemplos que muestran con detalle el comportamiento del programa OM de fusión de ontologías.

Primer ejemplo: fusión de ontologías que describen la novela Cien años de soledad, del escritor Gabriel García Márquez. En este caso las ontologías que representan una pequeña porción de la información de la novela

cuentan con más enlaces que conceptos, y esto se debe a que las relaciones o enlaces representan la descripción de los eventos, propiedades, hechos, de los personajes, lugares y cosas que intervienen en la narración. A continuación describimos brevemente los detalles de esta fusión. Se presenta un ejemplo donde se ha complementado la información de un nodo que está en ambas ontologías: se trata de la copia de las relaciones de un concepto. En la ontología A existe la afirmación “José Arcadio (nieto) murió en fusilamiento”; este hecho se representa en la relación “murió = José Arcadio (nieto), fusilamiento”; este hecho se ve más detalladamente en la ontología B, que dice “José Arcadio (nieto) murió en fusilamiento por los conservadores”; luego, en la ontología resultante aparece esta información más completa. (Las ontologías en este ejemplo se representan en la Figura 2, en notación OM; sólo se muestran las relaciones del concepto *Cien años de soledad* como una obra literaria.)

Segundo caso: fusión de ontologías que describen el estado de Oaxaca. En esta fusión se presenta la identificación de un concepto que en la ontología A es un nombre general (Oaxaca), mientras que en la ontología B es más específico (Oaxaca de Juárez, Oaxaca). Dada la cercanía de los conceptos “San Francisco Telixhuaca” y “Cuilapan de Guerrero”, así como “carretera número 131” en ambas ontologías, OM identifica que se trata del mismo concepto “Oaxaca”, que A refleja en B. Por tanto, OM busca la semántica

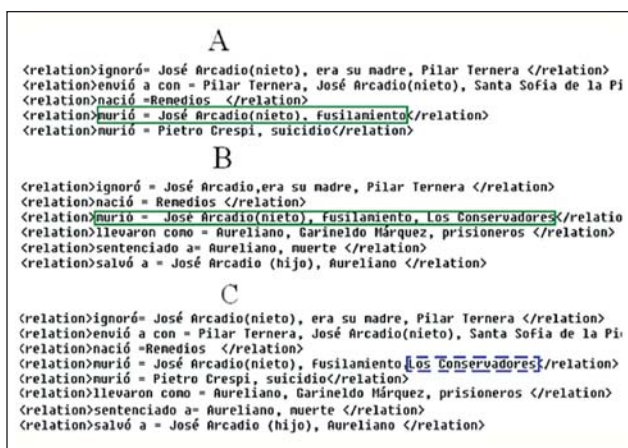


Figura 2. Ontología A y B (fuentes) donde coincide el nombre de una relación, pero B tiene información adicional; por tanto, C conjunta ambas, tanto información de A como de B.

entre “Oaxaca” y “Oaxaca de Juárez”, mediante una clasificación o taxonomía de lugares geográficos como herramienta interna. La búsqueda refleja que “Oaxaca de Juárez” es una ciudad del conjunto de ciudades del estado de Oaxaca. Por tanto, este nombre es más específico que “Oaxaca”, por lo que en la ontología resultante, C, queda el nombre específico en lugar del

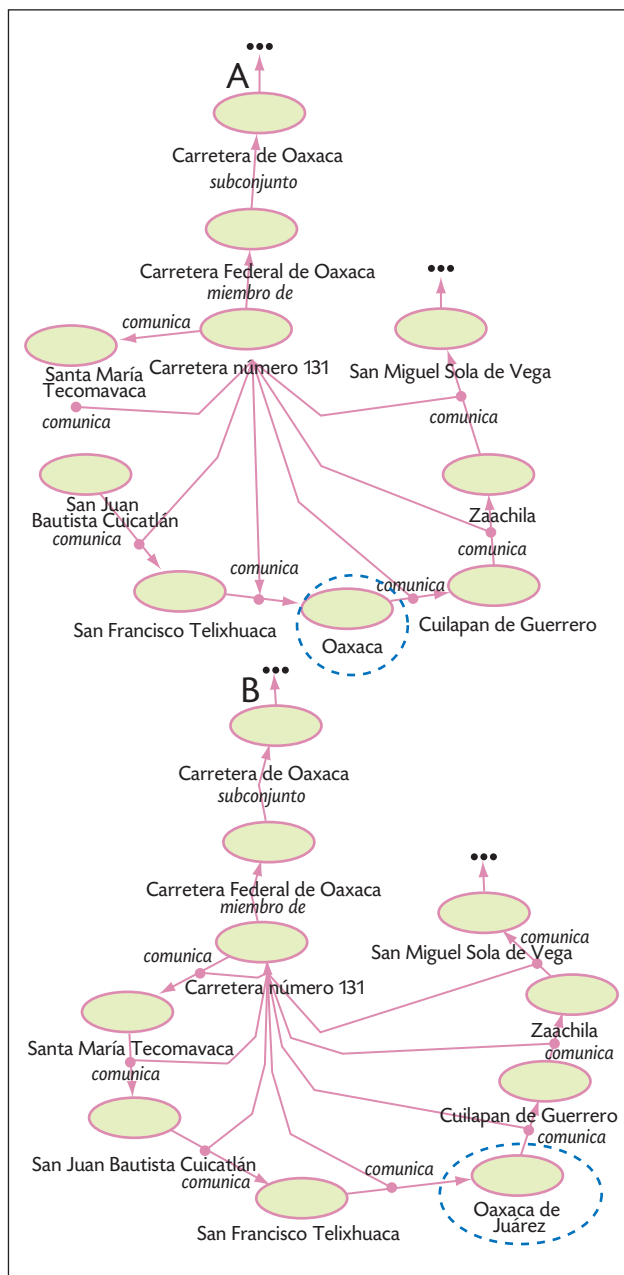


Figura 3. Ontologías A y B donde el concepto “Oaxaca” y “Oaxaca de Juárez” se identifican como nombres generales y particulares o específicos, respectivamente. Por tanto, la fusión lleva a la ontología resultante el concepto con el nombre más específico.

general. En la Figura 3 se presentan pequeñas porciones de las ontologías fuente de manera gráfica.

Tercer caso: *fusión de ontologías que describen las características de la amapola*. Éste es un caso de fusión con ontologías más pequeñas; obviamente, estas ontologías tienen un rango de error de 0 por ciento, comparadas con la fusión manual. Uno de los conflictos que se presentaron durante esta fusión se refiere a ciertas características que algunos conceptos no comparten. Por ejemplo, refiriéndose a las “amapolas”, se dice que estas flores “no tienen peciolo”; como se sabe, las amapolas tienen un tallo al final del cual descansa directamente la flor.

En la Figura 4 se presenta la ontología A con el concepto “hoja” y la relación “sin peciolo” (indicado en línea gruesa discontinua), mientras que en B existe el concepto “peciolo” (indicado en línea gruesa continua). OM verifica que el concepto “peciolo” no existe en la descripción de esta planta, por tanto, no copia este concepto. Por ello, en el resultado de la fusión, mostrado en la Figura 5, no aparece el concepto “peciolo”.

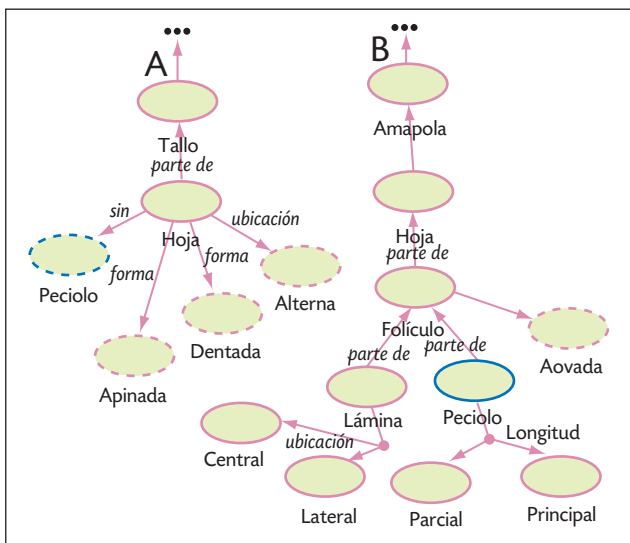


Figura 4. Ontologías fuente A y B; en la primera aparece que las amapolas no tienen peciolo (por espacio, no se muestra el concepto “amapola”, pues va arriba del concepto “tallo”), y en la segunda aparece que en el folículo de la amapola hay una parte que se llama peciolo. Realmente el fusionador “no descubre la verdad” sobre los hechos, sino que conserva como verdad la información más antigua. Lograr que el fusionador cambie de opinión según crea verdadera la información es un reto, pero se piensa implementar como una mejora.

Así como los anteriores ejemplos, existen muchos otros (no se muestran por limitaciones de espacio, pero pueden consultarse en Cuevas, 2006) que muestran el comportamiento OM y los detalles que resuelve de manera independiente, y que lo hacen ser automático. OM tiene así un grado importante de robustez que quizá hará posible que las computadoras puedan añadir por sí mismas información nueva a su “conocimiento”.

¿Una computadora puede aprender sola?

La respuesta a esta pregunta es afirmativa. Las computadoras aprenderán “fusionando” su información con la de otras computadoras. La información la pueden obtener de textos, escritos y documentos que ya se encuentran en Internet.

Existen billones de documentos en la red, bajo distintos esquemas o formatos, por ejemplo HTML (*HyperText Markup Language*, lenguaje de marcado de hipertexto), texto sin formato, gráficas, etcétera. Estos documentos se pueden convertir en ontologías de

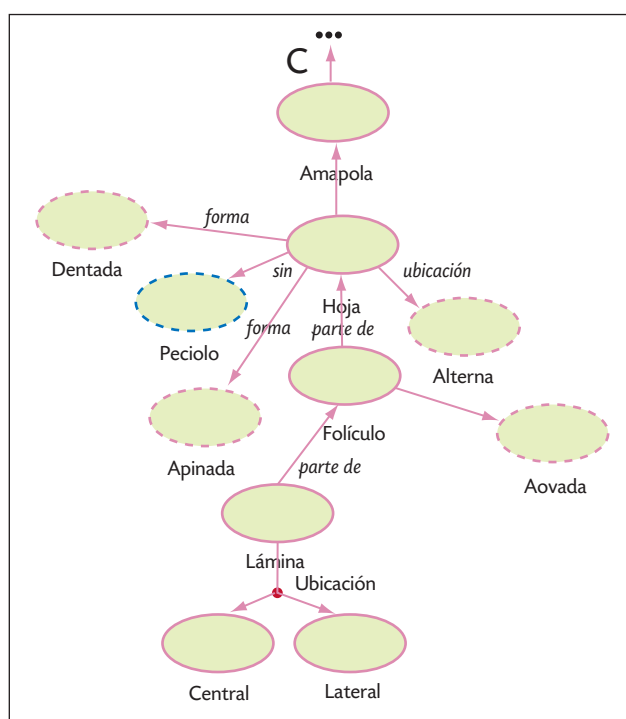


Figura 5. La ontología C resultante del ejemplo anterior (Figura 4). En ella se encuentra el concepto “peciolo” y no se han copiado los conceptos que estaban ligados a éste en la ontología B.

manera automática (esta conversión es parte de un nuevo proyecto, llamado OM que estamos empezando con mucho entusiasmo), y luego fusionarlas todas usando el fusionador, también automático. Así, se podrá tener una computadora o un grupo de computadoras que sabrán de muchas cosas, comunes y especializadas. Se les podrá hacer consultas y obtener respuestas, o localizar documentos más útiles al usuario.

Un buen ejemplo es el proyecto Wolfram Alpha (Wolfram Alpha Computational Knowledge Engine, www.wolframalpha.com), en el que se introduce una palabra, digamos “edad”, el programa ofrece la interpretación, su medida (tiempo), unidad estándar (año) y su sinónimo. Lo que deseamos hacer es un “sabelotodo” al que se le pueda preguntar cualquier cosa y nos dé la respuesta, en vez de tener que usar buscadores a los que les preguntamos algo vía palabras clave y nos señalan un montón de documentos que hay que procesar manualmente (leer, entender).

Conclusión y despedida

Con la invención de OM vendrán otros programas automáticos. Lo más urgente (ya estamos trabajando en eso con el proyecto OM*) es un analizador sintáctico-semántico que extraiga la información de un documento en español y la vierta a una ontología (construyéndola automáticamente, en notación OM).

Actualmente las ontologías que OM fusiona son hechas a mano, como ya se dijo. Este analizador hará uso de desambiguadores, lematizadores y otras herramientas de la lingüística computacional (en el CIC existe el Laboratorio de Lenguaje Natural y Procesamiento de Textos, que ha desarrollado algunas de estas herramientas). En el desarrollo de teorías relevantes, el análisis formal de conceptos (relativamente poco usado) ha sido complemento con la teoría de la confusión (Guzmán y Levachkine, 2004) y la Teoría de la Inconsistencia (Guzmán y Jiménez, 2010), ambas utilizadas por OM.

¿Qué tan lejos estamos de un “sabelotodo” que supere a Wolfram Alpha? Ya veremos dentro de unos tres años, cuando OM* esté concluyendo. Hasta entonces, hay mucho trabajo por hacer.

Alma Delia Cuevas Rasgado es miembro del Sistema Nacional de Investigadores y profesora de la Escuela Superior de Cómputo del Instituto Politécnico Nacional. Ha sido profesora en universidades del gobierno federal y privadas, y consultora y diseñadora de sistemas en instituciones públicas como la Dirección General de Educación Superior Tecnológica y la Secretaría de Educación Pública de Hidalgo. Escribió un libro de diseño de sistemas orientados a procesos (actualmente en revisión) en la empresa mexicana Servicios Corporativos ITComplements, así como el capítulo sobre tecnologías de información (en revisión por el comité de la editorial) en un libro de texto para bachillerato de la editorial McMillan de México. Sus áreas de interés son la ingeniería de *software* y la inteligencia artificial, el procesamiento semántico, las ontologías, la fusión del conocimiento y la deducción de información no trivial contenida en documentos.

almadeliacuevas@gmail.com

Bibliografía

- Cuevas, A. (2006), “Unión de ontologías usando propiedades semánticas”, tesis de doctorado, CIC-IPN, México (disponible en: www.divshare.com/download/6096305-b18).
- Cycorp (2002), *Ontological engineer's handbook* (disponible en: www.cyc.com/doc/handbook/oe/oe-handbook-toc-opencyc.html).
- Dou, D., D. McDermott y P. Qi (2002), “Ontology translation by ontology merging and automated reasoning”, *Proceedings of the EKAW Workshop on Ontologies for Multi-agent Systems*.
- Fridman, N. y M. Musen (2000), “PROMPT: algorithm and tool for automated ontology merging and alignment”, *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, Austin, Texas, pp. 450-455.
- Guzmán, A. y A. Jiménez (2010), “Obtaining the consensus and inconsistency among a set of assertions on a qualitative attribute”, *Expert systems with applications*; 37: 158-164.
- Guzmán, A. y S. Levachkine (2004), “Hierarchies measuring qualitative variables”, *Computational linguistics and intelligent text processing* (serie Lecture notes in computer science), Springer-Verlag.
- Kotis, K., G. Vouros y K. Stergiou (2005), “Towards automatic of domain ontologies: the HCONE-merge approach”, *Elsevier's journal of web semantics*; 4(1): 60-79 (disponible en: <http://authors.elsevier.com/sd/article/S1570826805000259>).
- Stumme, G. y A. Maedche (2002), “Ontology merging for federated ontologies on the semantic web”, en E. Franconi, K. Barker y D. Calvanese (eds.), *Proceedings of the International Workshop on Foundations of Models for Information Integration (FMII'01)*, Viterbo, Italia, LNAI, Springer.

