Lenguajes y modelos subyacentes a los grafos de conocimiento

Renzo Angles
rangles@utalca.cl
https://orcid.org/0000-0002-6740-9711
Universidad de Talca, Chile

Recibido: 7 de septiembre del 2022 / Aceptado: 10 de octubre del 2022
doi: https://doi.org/10.26439/ciis2022.6065

RESUMEN. Un grafo de conocimiento es una gran base de datos que integra información desde distintas fuentes de datos, con el objetivo de poder extraer conocimiento y transformarlo en valor para los usuarios. Esta base de datos es representada como un grafo, donde los nodos representan entidades, y cada arista representa una relación entre dos nodos o un atributo de un nodo. El objetivo de este artículo es presentar una revisión de los modelos de datos que se usan para representar grafos de conocimiento, y los lenguajes de consulta que permiten extraer la información explícita e implícita contenida en dichos grafos.

PALABRAS CLAVE: grafo de conocimiento, modelos de datos basados en grafos, lenguaje de consulta para grafos

LANGUAGES AND MODELS UNDERLYING KNOWLEDGE GRAPHS

ABSTRACT. A knowledge graph is a large database that integrates information from different data sources, this with the objective of supporting knowledge extraction, and allowing the transformation of such knowledge into value for the users. Such a database is represented as a graph where the nodes represent entities, and each edge represents a relationship between nodes, or the attribute of a node. The objective of this article is to review the data models used to represent knowledge graphs, and the query languages that allow to extract explicit and implicit information contained in such graphs.

KEYWORDS: knowledge graph, graph data model, graph query language

1. INTRODUCCIÓN

Los términos *dato*, *información* y *conocimiento* son fundamentales en áreas del conocimiento de gran interés científico e impacto tecnológico en la actualidad. Entre dichas áreas se encuentran bases de datos (databases), datos masivos (*big data*), web semántica (*semantic web*), inteligencia artificial, entre otros.

Para explicar dichos conceptos, consideremos la siguiente frase: "Rosa Rosales cortó una rosa, que roja es la rosa de Rosa Rosales".

En términos muy generales, un dato puede definirse como una palabra sin significado obvio, o una palabra ambigua (es decir, puede entenderse o interpretarse de diversas maneras). Por ejemplo, la palabra *rosa* puede tener varios significados: una flor, un nombre propio, un color, una marca o incluso un lugar. Ahora, si acotamos dichos significados a la frase del ejemplo, entonces, *rosa* puede ser un nombre o una flor. Luego, para determinar el significado preciso de una ocurrencia de la palabra *rosa* en la frase del ejemplo, debemos analizar las palabras que aparecen a su alrededor. Por ejemplo, si consideramos el fragmento "Rosa Rosales", podemos inferir que *Rosa* es un nombre propio; en el caso de "que roja es la rosa", podemos saber que se refiere a una flor. Es decir, el significado de un dato puede dilucidarse analizando las relaciones que tiene con otros datos. En este sentido, la información se puede definir como un conjunto de datos cuyo significado se infiere de sus relaciones y un contexto. Finalmente, el término *conocimiento* puede definirse como la información implícita que puede extraerse al procesar la información explícita o existente. Por ejemplo, el número de veces que cada palabra aparece y la palabra que aparece el mayor número de veces son dos ejemplos de conocimiento que pueden extraerse de la frase del ejemplo.

Desde el inicio de la era digital, a finales de los años cincuenta, se han investigado problemas y desarrollado soluciones para gestionar datos, información y conocimiento. Inicialmente, la preocupación estaba en almacenar (codificar) y procesar (leer y escribir) los datos. Con la aparición del modelo de datos relacional en 1970, se propusieron distintas formas de modelar (o representar) los datos y extraer (o consultar) la información subyacente. La creación de la Web (en 1989) marca el inicio de otra etapa, una donde las personas y los sistemas empiezan a generar información y conocimiento gracias a estándares para transferir datos (HTTP), generar información (HTML) y representar conocimiento (RDF, RDF Schema, OWL). La Web motiva el desarrollo de sistemas informáticos interconectados, los cuales originan información que se caracteriza por su volumen (cantidad), variedad (heterogeneidad) y velocidad (de generación), dando lugar al concepto de *big data*. En este punto, se empiezan a desarrollar plataformas (como Apache Hadoop) que permiten el procesamiento y análisis de datos masivos, empleando técnicas de almacenamiento distribuido de datos y computación paralela. Las organizaciones se dan cuenta del conocimiento implícito existente en los datos que producen, por lo que empiezan a impulsar proyectos de ciencia de datos (*data science*) con el objetivo de generar valor empleando métodos estadísticos y técnicas avanzadas de minería

de datos. La existencia de grandes cantidades de datos no solo permite extraer conocimiento, sino también que los sistemas de inteligencia artificial aprendan de los datos a fin de lograr predicciones más precisas. Podríamos decir que actualmente nos encontramos en "la era del conocimiento"', ya que estamos desarrollando métodos para representar, analizar y generar conocimiento, o mejor dicho, grafos de conocimiento.

2. GRAFOS DE CONOCIMIENTO

No existe una definición estándar para el término *grafo de conocimiento* (*knowledge graph*), pero presentaremos tres definiciones que consideramos relevantes:

- Un grafo dirigido, cuyos nodos son unidades de conocimiento (conceptos) que un estudiante debe adquirir, y cuyas aristas denotan dependencias entre dichas unidades de conocimiento (Schneider, 1973). Esta es la primera definición del término *grafo de conocimiento*, que se incluye en un artículo publicado en 1973.
- Un modelo inteligente (un grafo) que comprende entidades del mundo real y las relaciones entre ellas. Esto corresponde a un fragmento del *post* que se usó para dar a conocer el Google Knowledge Graph (Singhal, 2012).
- Un grafo de datos destinado a acumular y transmitir conocimiento del mundo real, cuyos nodos representan entidades de interés, y cuyas aristas representan relaciones potencialmente diferentes entre dichas entidades. Esta definición es parte de un artículo (Hogan et al., 2021) que revisa modelos, lenguajes, herramientas y dominios de aplicación asociados a los grafos de conocimiento.

Luego de revisar diversos trabajos en el área, nuestra definición de grafo de conocimiento es la siguiente: una gran base de datos que integra información desde distintas fuentes de datos con el objetivo de generar información adicional y conocimiento. Dicha base de datos es representada como un grafo; es decir, las entidades se representan como nodos, y las relaciones entre dichas entidades se representan como aristas.

Actualmente existen diversos modelos, lenguajes y métodos que nos permiten representar, analizar (consultar) y generar grafos de conocimiento. A continuación, describimos algunos de ellos.

3. MODELOS PARA REPRESENTAR GRAFOS DE CONOCIMIENTO

El desarrollo de un grafo de conocimiento es un proceso complejo. Una de las primeras tareas es el modelado conceptual, que consiste en identificar tipos de entidades, tipos de relaciones y tipos de restricciones, y representarlos usando un modelo de datos (Brodie et al., 1984). De

una manera muy general, un modelo de datos se define como una colección de herramientas conceptuales usadas para modelar representaciones de entidades del mundo real y las relaciones entre ellas (Silberschatz et al., 1996). Existen distintos modelos de datos (Beeri, 1988), los cuales pueden agruparse según la estructura abstracta en que están basados, por ejemplo, tablas, árboles o grafos.

El modelado conceptual de un grafo de conocimiento puede realizarse usando cualquier modelo de datos, pero es más natural y usual emplear un modelo basado en grafos, ya que está pensado para representar de mejor manera las conexiones entre las entidades. Existen diversos modelos de datos basados en grafos, algunos teóricos (Angles & Gutierrez, 2008) y otros más prácticos (Angles et al., 2017). Los modelos más usados en la actualidad son tres: grafo dirigido etiquetado, grafo con propiedades y grafo RDF.

Un grafo dirigido etiquetado (*directed labeled graph*) (Barceló Baeza, 2013) es una estructura compuesta de nodos y aristas, donde los nodos y las aristas pueden tener identificadores y etiquetas; cada arista conecta un par de nodos; las aristas son dirigidas, ya que tienen un nodo origen y destino; y pueden existir múltiples aristas entre dos nodos (multigrafo). Desde el punto de vista de modelado de datos, un nodo representa una entidad o un valor, y una arista representa una relación entre dos entidades o un atributo de una entidad. Este modelo es muy usado en métodos estadísticos, minería de datos, inteligencia artificial y sistemas de procesamiento masivo de datos.

Un grafo con propiedades (*property graph*) (Angles, 2018) es un grafo dirigido etiquetado, pero tiene una característica extra: cada nodo o arista puede mantener un conjunto (posiblemente vacío) de propiedades, donde una propiedad tiene un nombre (o etiqueta) y un valor. En este modelo, un nodo representa una entidad, una arista representa una relación entre dos entidades, y una propiedad representa una característica específica y propia de una entidad o una relación. Este modelo es muy usado por los sistemas de gestión de bases de datos que soportan el almacenamiento y consulta de grafos (Angles & Gutierrez, 2018).

RDF (*resource description framework*) es un estándar para describir recursos web, que define un modelo de datos basado en grafos. En un grafo RDF existen tres tipos de nodos: recursos web identificados por una especie de URL (llamada IRI), nodos blancos que representan recursos anónimos, y literales que representan datos o valores concretos (como cadenas, números y fechas). Un triple RDF es una tupla de la forma (sujeto, predicado, objeto), donde el sujeto puede ser un recurso web o un recurso anónimo; el predicado suele ser un recurso web (que referencia a un tipo de relación), y el objeto puede ser un recurso web o un valor concreto. Según lo anterior, el sujeto y el objeto serían nodos en el grafo, y el predicado es el identificador de una relación. Nótese que, en comparación con un grafo con propiedades, una arista puede representar una relación o una propiedad. Un grafo RDF es la manera estándar de representar datos en la web semántica.

Existen otros modelos que tratan de sobrellevar las deficiencias de estos modelos (Angles et al., 2022), o han sido diseñados para dominios de aplicación especiales. Por ejemplo, el modelo basado en hipergrafos (*hypergraphs*) (Iordanov, 2010) permite representar relaciones n-arias con más facilidad, pero su implementación es más compleja. RDF* (Hartig, 2017) extiende el modelo RDF con el objetivo de permitir metadatos sobre las propiedades de un triple RDF. En un trabajo reciente (Lassila et al., en prensa), los autores plantean la idea de un único modelo de datos basado en grafos denominado *OneGraph*, el cual busca unificar RDF y los grafos con propiedades con el objetivo de permitir interoperabilidad entre ambos modelos y facilitar el desarrollo de lenguajes de consulta.

Actualmente existen diversos sistemas de gestión de datos que permiten almacenar y consultar grafos de conocimiento (DB-Engines, 2022). Estos sistemas se pueden dividir en cuatro grupos: sistemas de base de datos para grafos (*graph database systems*), como Neo4j (http://neo4j.com/) y TigerGraph (https://www.tigergraph.com), que en su mayoría soportan *property graphs*; RDF *triple stores*, como Apache Jena (https://jena.apache.org/) y GraphDB (https://graphdb.ontotext.com), que son diseñados para gestionar grafos RDF; plataformas de procesamiento distribuido de grafos (*distributed graph processing frameworks*), como Apache Giraph (http://giraph.apache.org) y GraphX (https://spark.apache.org/graphx/), que permiten manipular grafos dirigidos etiquetados de gran tamaño; y sistemas multimodelo, como Amazon Neptune (https://aws.amazon.com/es/neptune/) y Microsoft Azure Cosmos DB (https://docs.microsoft.com/en-us/azure/cosmos-db/), que soportan múltiples modelos, usualmente RDF y grafos con propiedades.

4. LENGUAJES PARA CONSULTAR GRAFOS DE CONOCIMIENTO

Un componente clave de cualquier sistema de gestión de datos es su lenguaje de consulta. En términos generales, un lenguaje de consulta es un lenguaje computacional de alto nivel que permite recuperar los datos almacenados en un sistema de gestión de datos (Samet, 1981). En el caso de los sistemas de gestión de datos basados en grafos, un lenguaje de consulta para grafos (*graph query language*) está diseñado para extraer información usando operaciones orientadas a grafos, como patrones y consultas de caminos (Angles, Reutter & Voigt, 2018).

Existen varios trabajos relacionados con los lenguajes de consulta para grafos, los cuales tratan aspectos como su definición (Angles & Gutierrez, 2008; Angles, 2012), expresividad (es decir, los tipos de consultas que el usuario puede expresar) (Angles et al., 2022), complejidad computacional (esto es, el tiempo requerido para evaluar distintos tipos de consultas) (Barceló Baeza, 2013; Angles et al., 2017; Bonifati & Dumbrava, 2019), implementación (Fletcher et al., 2016) y evaluación (Ciglan et al., 2012). El libro de Bonifati et al. (2018) presenta una revisión completa sobre consultas orientadas a grafos.

A pesar de los avances en el desarrollo de lenguajes de consulta para grafos, aún no existe un estándar. Sin embargo, en septiembre del 2019, se inició el proyecto GQL Standard (https://www.gqlstandards.org/) cuyo objetivo es definir un lenguaje estándar que se basa en la fusión de cuatro lenguajes existentes: Cypher (Neo4j) (Cypher Query Language, s. f.), PGQL (Oracle) (Van Rest et al., 2013), G-CORE (LDBC) (Angles, Arenas et al., 2018) y GSQL (TigerGraph) (Deutsch et al., 2020).

La principal noción detrás de un lenguaje de consulta para grafos es la búsqueda de patrones (*graph pattern matching*) (Gallagher, 2006). Un patrón de grafo simple es un subgrafo donde los nodos y las aristas pueden ser entidades, relaciones y variables. Los patrones simples se pueden combinar usando operadores relacionales, como reunión (*join*), unión y diferencia, permitiendo definir patrones más complejos (Angles et al., 2022). El resultado de un patrón de grafo puede ser un conjunto de grafos que satisfacen la estructura del patrón, o una tabla donde cada columna es una variable y cada fila contiene asignaciones de variables a nodos o valores.

Aunque muchas consultas reales pueden ser expresadas como patrones de grafos, es muy frecuente necesitar de mecanismos adicionales que permitan explorar la topología del grafo (Angles et al., 2022). Esto nos lleva a la noción de consulta de camino (*path query*) (Barceló et al., 2012), que tiene que ver con navegar a través del grafo usando patrones de camino. Recordemos que, en la teoría de grafos, un camino es una secuencia de nodos y aristas que nos permiten ir desde un nodo origen a un nodo destino (Diestel, 2005). En este sentido, la forma más común de representar consultas de caminos es a través de una expresión de la forma (a,p,b), donde *a* representa el nodo origen, *b* el nodo destino, y *p* es una expresión regular (Mendelzon & Wood, 1995). Los aspectos teóricos asociados a las consultas de caminos se han estudiado de manera amplia (Angles et al., 2022; Bonifati et al., 2018), y los sistemas de gestión de bases de datos soportan diversos tipos de consultas de caminos. Sin embargo, aún persiste el desafío de ejecutar eficientemente consultas de caminos en grafos de gran tamaño, debido a la complejidad computacional intrínseca del problema.

Si bien los lenguajes de consulta para grafos permiten extraer información desde los grafos de conocimiento, no son lo suficientemente poderosos para expresar consultas complejas (conocimiento) propias del análisis de grafos (*graph analytics*). Por esta razón, se han desarrollado lenguajes que mezclan operaciones declarativas con programación, como Gremlin (Apache TinkerPop, s. f.); librerías que proveen algoritmos para grafos, como The Neo4j Graph Data Science Library (Neo4j, 2021); y sistemas especiales que permiten procesamiento de grafos a gran escala, como Spark GraphX (https://spark.apache.org/graphx/).

5. CONCLUSIONES

Actualmente las organizaciones están creando muchos grafos de conocimiento con el objetivo de obtener valor de sus datos. Sin bien existen modelos y lenguajes bien establecidos para representar los grafos de conocimiento, aún hay varios desafíos prácticos relacionados con la consulta y análisis de estos grafos, si se piensa en extraer de manera eficiente el conocimiento oculto.

Además de representar y consultar los datos, necesitamos modelos para representar el conocimiento, así como métodos deductivos e inductivos que permitan generar conocimiento de manera automática. Estos temas son el foco de investigación actual en el área y representan el estado actual de la era del conocimiento.

REFERENCIAS

Angles, R. (2012). A comparison of current graph database models. En *4rd International Workshop on Graph Data Management: Techniques and Applications (GDM) (ICDE Workshop)*.

Angles, R. (2018). The property graph database model. En *Proceedings of the Alberto Mendelzon International Workshop on Foundations of Data Management (AMW)* (vol. 2100). CEUR Workshop Proceedings.

Angles, R., Arenas, M., Barceló, P., Boncz, P., Fletcher, G., Gutierrez, C., ... Voigt, H. (2018). G-CORE: A core for future Graph Query languages. En *Proceedings of the International Conference on Management of Data (SIGMOD)*. ACM.

Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., & Vrgoč, D. (2017). Foundations of modern query languages for graph databases. *ACM Computing Surveys*, *50*(5).

Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys*, *40*(1), 1-39.

Angles, R., & Gutierrez, C. (2018). An introduction to graph data management. En *Graph data management* (cap. 1). Springer Nature.

Angles, R., Hogan, A., Lassila, O., Rojas, C., Schwabe, D., Szekely, P., & Vrgoč, D. (2022). Multilayer graphs: A unified data model for graph databases. En *Proceedings of the 5th ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*. ACM.

Angles, R., Reutter, J., & Voigt, H. (2018). Graph query languages. En *Encyclopedia of big data technologies* (pp. 1-8). Springer International Publishing.

Apache TinkerPop. (s. f.). *Gremlin Query Language*. https://tinkerpop.apache.org/gremlin.html

Barceló, P., Libkin, L., Lin, A. W., & Wood, P. T. (2012). Expressive languages for path queries over graph-structured data. *ACM Transactions on Database Systems*, *37*(4), 1-46.

Barceló Baeza, P. (2013). Querying graph databases. En *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)* (pp. 175-188). ACM. https://doi.org/10.1145/2463664.2465216

Beeri, C. (1988). Data models and languages for databases. En *Proceedings of the 2nd International Conference on Database Theory (ICDT)* (vol. 326, pp. 19-40). Springer.

Bonifati, A., & Dumbrava, S. (2019). Graph queries: From theory to practice. *SIGMOD Record*, *47*(4), 5-16.

Bonifati, A., Fletcher, G., Voigt, H., & Yakovets, N. (2018). *Querying graphs*. Morgan & Claypool Publishers.

Brodie, M. L., Mylopoulos, J., & Schmidt, J. W. (1984). *On conceptual modelling*. Springer-Verlag.

Ciglan, M., Averbuch, A., & Hluchy, L. (2012). Benchmarking traversal operations over graph databases. En *Proceedings of the International Conference on Data Engineering Workshops* (pp. 186-189). IEEE Computer Society.

Cypher Query Language. (s. f.). http://neo4j.com/developer/cypher-query-language/.

DB-Engines. (2022, noviembre). *DB-Engines Ranking of Graph DBMS*. http://db-engines.com/en/ranking/graph+dbms.

Deutsch, A., Xu, Y., Wu, M., & Lee, V. E. (2020). Aggregation support for modern graph analytics in TigerGraph. En *Proceedings of the International Conference on Management of Data (SIGMOD)* (pp. 377-392). ACM.

Diestel, R. (2005). *Graph theory* (3.ª ed., vol. 173). Springer-Verlag.

Fletcher, G. H. L., Peters, J., & Poulovassilis, A. (2016). Efficient regular path query evaluation using path indexes. En *Proceedings of the 19th International Conference on Extending Database Technology* (pp. 636-639). OpenProceedings.org.

Gallagher, B. (2006). Matching structure and semantics: A survey on graph-based pattern matching. En *AAAI Fall Symposium on Capturing and Using Patterns for Evidence Detection* (pp. 45-53). The AAAI Press.

Hartig, O. (2017). Foundations of RDF* and SPARQL* - An alternative approach to statement-level metadata in RDF. En *Proceedings of the 11th Alberto Mendelzon*

International Workshop on Foundations of Data Management and the Web (vol. 1912). CEUR Workshop Proceedings.

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. D., Gutierrez, C., ... Zimmermann, A. (2021). Knowledge graphs. *ACM Computing Surveys*, *54*(4).

Iordanov, B. (2010). HyperGraphDB: A generalized graph database. En *Web-Age Information Management. WAIM 2010. Lecture Notes in Computer Science* (vol. 6185, pp. 25-36). Springer-Verlag.

Lassila, O., Schmidt, M., Hartig, O., Bebee, B., Bechberger, D., Broekema, W., ... Thompson, B. (en prensa). The OneGraph vision: Challenges of breaking the graph model lock-in. *Semantic Web*, *14*.

Mendelzon, A. O., & Wood, P. T. (1995). Finding regular simple paths in graph databases. *SIAM Journal on Computing*, *24*(6), 1235-1258.

Neo4j. (2022). *The Neo4j Graph Data Science Library*. https://neo4j.com/docs/graph-data-science/current/

Samet, J. (Ed.). (1981). *Query languages. A unified approach. Report of the British Computer Society Query Languages Group*. Heyden University Press.

Schneider, E. W. (1973). *Course modularization applied: The interface system and its implications for sequence control and data analysis*. https://files.eric.ed.gov/fulltext/ED088424.pdf

Silberschatz, A., Korth, H. F., & Sudarshan, S. (1996). Data models. *ACM Computing Surveys*, *28*(1), 105-108.

Singhal, A. (2012, 16 de mayo). Introducing the Knowledge Graph: things, not strings. *The Keyword*. https://blog.google/products/search/introducing-knowledge-graph-things-not/

Van Rest, O., Hong, S., Kim, J., Meng, X., & Chafi, H. (2016). PGQL: A property Graph Query Language. En *Proceedings of the Fourth International Workshop on Graph Data Management Experiences and Systems*.