WIKIPEDIA
The Free Encyclopedia

Knowledge graph

In knowledge representation and reasoning, a **knowledge graph** is a knowledge base that uses a graph-structured data model or topology to represent and operate on data. Knowledge graphs are often used to store interlinked descriptions of entities – objects, events, situations or abstract concepts – while also encoding the semantics or relationships underlying these entities.[1]

Example conceptual diagram

Since the development of the Semantic Web, knowledge graphs have often been associated with linked open data projects, focusing on the connections between concepts and entities.[2][3] They are also historically associated with and used by search engines such as Google, Bing, Yext and Yahoo; knowledge-engines and question-answering services such as WolframAlpha, Apple's Siri, and Amazon Alexa; and social networks such as LinkedIn and Facebook.

Recent developments in data science and machine learning, particularly in graph neural networks and representation learning, have broadened the scope of knowledge graphs beyond their traditional use in search engines and recommender systems. They are increasingly used in scientific research, with notable applications in fields such as genomics, proteomics, and systems biology.[4]

History

The term was coined as early as 1972 by the Austrian linguist Edgar W. Schneider, in a discussion of how to build modular instructional systems for courses.[5] In the late 1980s, the University of Groningen and University of Twente jointly began a project called Knowledge Graphs, focusing on the design of semantic networks with edges restricted to a limited set of relations, to facilitate algebras on the graph. In subsequent decades, the distinction between semantic networks and knowledge graphs was blurred.

Some early knowledge graphs were topic-specific. In 1985, Wordnet was founded, capturing semantic relationships between words and meanings – an application of this idea to language itself. In 2005, Marc Wirk founded Geonames to capture relationships between different geographic names and locales and associated entities. In 1998 Andrew Edmonds of Science in Finance Ltd in the UK created a system called ThinkBase that offered fuzzy-logic based reasoning in a graphical context.[6]

In 2007, both DBpedia and Freebase were founded as graph-based knowledge repositories for general-purpose knowledge. DBpedia focused exclusively on data extracted from Wikipedia, while Freebase also included a range of public datasets. Neither described themselves as a 'knowledge graph' but developed and described related concepts.

In 2012, Google introduced their Knowledge Graph,[7] building on DBpedia and Freebase among other sources. They later incorporated RDFa, Microdata, JSON-LD content extracted from indexed web pages, including the *CIA World Factbook*, Wikidata, and Wikipedia.[7][8] Entity and relationship types associated with this knowledge graph have been further organized using terms
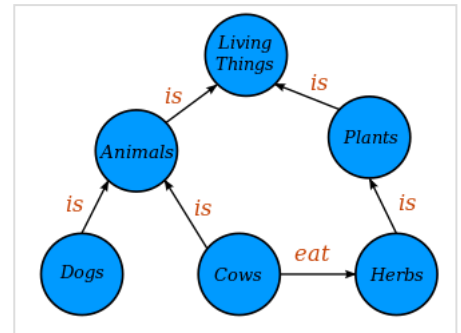
from the schema.org[9] vocabulary. The Google Knowledge Graph became a successful complement to string-based search within Google, and its popularity online brought the term into more common use.[9]

Since then, several large multinationals have advertised their knowledge graphs use, further popularising the term. These include Facebook, LinkedIn, Airbnb, Microsoft, Amazon, Uber and eBay.[10]

In 2019, IEEE combined its annual international conferences on "Big Knowledge" and "Data Mining and Intelligent Computing" into the International Conference on Knowledge Graph.[11]

Definitions

There is no single commonly accepted definition of a knowledge graph. Most definitions view the topic through a Semantic Web lens and include these features:[12]

- *Flexible relations among knowledge in topical domains*: A knowledge graph (i) defines abstract classes and relations of entities in a schema, (ii) mainly describes real world entities and their interrelations, organized in a graph, (iii) allows for potentially interrelating arbitrary entities with each other, and (iv) covers various topical domains.[13]
- *General structure*: A network of entities, their semantic types, properties, and relationships.[14][15] To represent properties, categorical or numerical values are often used.
- *Supporting reasoning over inferred ontologies*: A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge.[2]

There are, however, many knowledge graph representations for which some of these features are not relevant. For those knowledge graphs, this simpler definition may be more useful:

- A digital structure that represents knowledge as concepts and the relationships between them (facts). A knowledge graph can include an ontology that allows both humans and machines to understand and reason about its contents.[16][17]

Implementations

In addition to the above examples, the term has been used to describe open knowledge projects such as YAGO and Wikidata; federations like the Linked Open Data cloud;[18] a range of commercial search tools, including Yahoo's semantic search assistant Spark, Google's Knowledge Graph, and Microsoft's Satori; and the LinkedIn and Facebook entity graphs.[2]

The term is also used in the context of note-taking software applications that allow a user to build a personal knowledge graph.[19]

The popularization of knowledge graphs and their accompanying methods have led to the development of graph databases such as Neo4j[20] and GraphDB.[21] These graph databases allow users to easily store data as entities their interrelationships, and facilitate operations such as data reasoning, node embedding, and ontology development on knowledge bases.

Using a knowledge graph for reasoning over data

A knowledge graph formally represents semantics by describing entities and their relationships.[22] Knowledge graphs may make use of ontologies as a schema layer. By doing this, they allow logical inference for retrieving implicit knowledge rather than only allowing queries requesting explicit
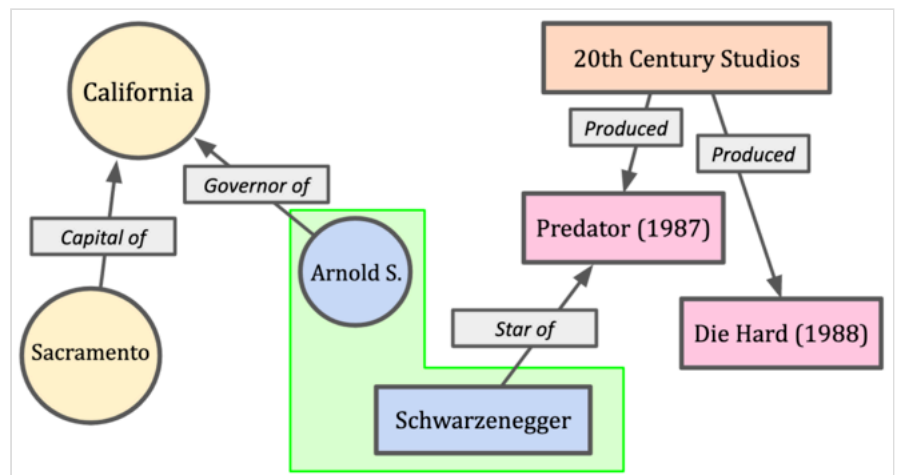
knowledge.[23]

In order to allow the use of knowledge graphs in various machine learning tasks, several methods for deriving latent feature representations of entities and relations have been devised. These knowledge graph embeddings allow them to be connected to machine learning methods that require feature vectors like word embeddings. This can complement other estimates of conceptual similarity.[24][25][26]

Models for generating useful knowledge graph embeddings are commonly the domain of graph neural networks (GNNs).[27] GNNs are deep learning architectures that comprise edges and nodes, which correspond well to the entities and relationships of knowledge graphs. The topology and data structures afforded by GNNS provides a convenient domain for semi-supervised learning, wherein the network is trained to predict the value of a node embedding (provided a group of adjacent nodes and their edges) or edge (provided a pair of nodes). These tasks serve as fundamental abstractions for more complex tasks such as knowledge graph reasoning and alignment.[28]

Entity alignment

As new knowledge graphs are produced across a variety of fields and contexts, the same entity will inevitably be represented in multiple graphs. However, because no single standard for the construction or representation of knowledge graph exists, resolving which entities from disparate graphs correspond to the same real world subject is a non-trivial task. This task is known as *knowledge graph entity alignment*, and is an active area of research.[29]

Two hypothetical knowledge graphs representing disparate topics contain a node that corresponds to the same entity in the real world. Entity alignment is the process of identifying such nodes across multiple graphs.

Strategies for entity alignment generally seek to identify similar substructures, semantic relationships, shared attributes, or combinations of all three between two distinct knowledge graphs. Entity alignment methods use these structural similarities between generally non-isomorphic graphs to predict which nodes corresponds to the same entity.[30]

The recent successes of large language models (LLMs), in particular their effectiveness at producing syntactically meaningful embeddings, has spurred the use of LLMs in the task of entity alignment.[31]