



The heart of the knowledge graph is a knowledge model – a collection of interlinked descriptions of concepts, entities, relationships and events where:

- Descriptions have formal semantics that allow both people and computers to process them in an efficient and unambiguous manner;
- Descriptions contribute to one another, forming a network, where each entity represents part of the description of the entities related to it;
- Diverse data is connected and described by semantic metadata according to the knowledge model.

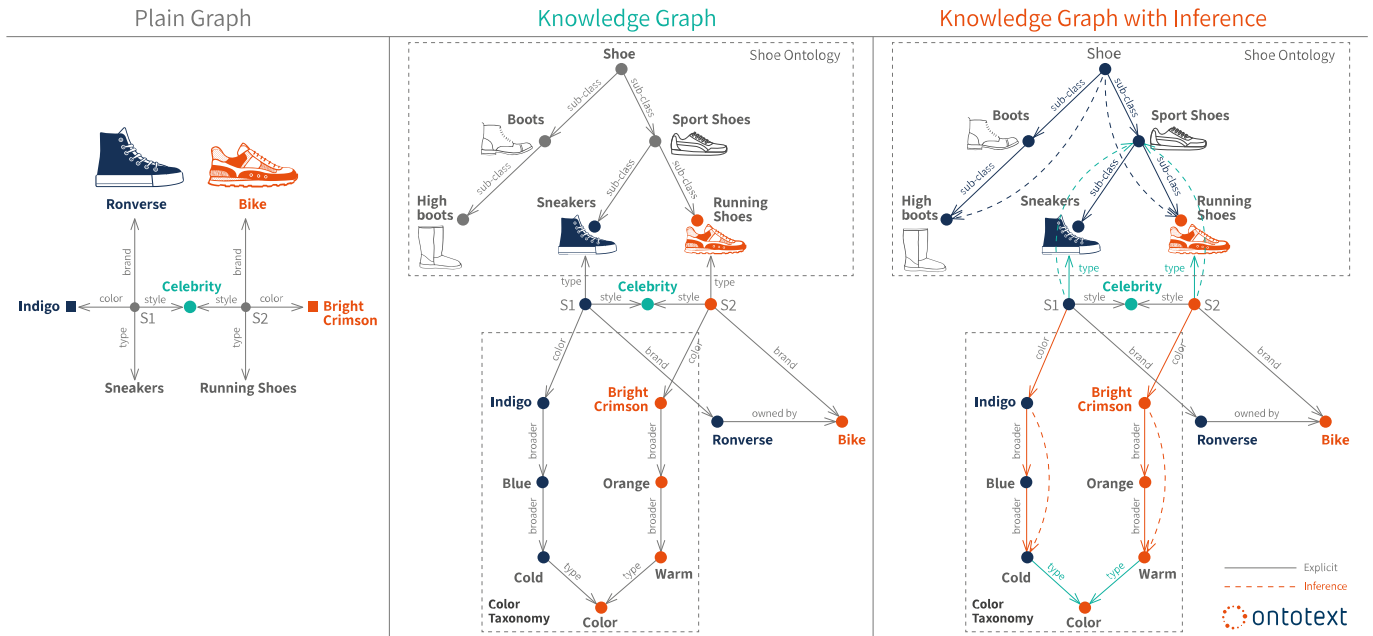
Key Characteristics

Knowledge graphs combine characteristics of several data management paradigms:

- **Database**, because the data can be explored via structured queries;
- **Graph**, because they can be analyzed as any other network data structure;
- **Knowledge base**, because they bear formal semantics, which can be used to interpret the data and infer new facts.

Knowledge graphs, represented in [RDF](#), provide the best framework for data integration, unification, linking and reuse, because they combine:

- **Expressivity**: The standards in the Semantic Web stack – RDF(S) and OWL – allow for a fluent representation of various types of data and content: data schema, taxonomies and vocabularies, all sorts of [metadata](#), reference and master data. The [RDF*](#) extension makes it easy to model provenance and other structured metadata.
- **Performance**: All the specifications have been thought out, and proven in practice, to allow for efficient management of graphs of billions of facts and properties.
- **Interoperability**: There is a range of specifications for data serialization, access (SPARQL Protocol for end-points), management (SPARQL Graph Store) and federation. The use of globally unique identifiers facilitates data integration and publishing.
- **Standardization**: All the above is standardized through the W3C community process, to make sure that the requirements of different actors are satisfied – all the way from logicians to enterprise data management professionals and system operations teams.



Click on image to enlarge

Ontologies and Formal Semantics

Ontologies represent the backbone of the formal semantics of a knowledge graph. They can be seen as the data schema of the graph. They serve as a formal contract between the developers of the knowledge graph and its users regarding the meaning of the data in it. A user could be another human being or a software application that wants to interpret the data in a reliable and precise way. Ontologies ensure a shared understanding of the data and its meanings.

When formal semantics are used to express and interpret the data of a knowledge graph, there are a number of representation and modeling instruments:

- **Classes.** Most often an entity description contains a classification of the entity with respect to a class hierarchy. For instance, when dealing with business information there could be classes *Person*, *Organization* and *Location*. Persons and organizations can have a common superclass *Agent*. Location usually has numerous sub-classes, e.g., *Country*, *Populated place*, *City*, etc. The notion of class is borrowed by the object-oriented design,



where each entity usually belongs to exactly one class.

- **Relationship types.** The relationships between entities are usually tagged with types, which provide information about the nature of the relationship, e.g., *friend*, *relative*, *competitor*, etc. Relationship types can also have formal definitions, e.g., that *parent-of* is inverse relation of *child-of*, they both are special cases of *relative-of*, which is a symmetric relationship. Or defining that *sub-region* and *subsidiary* are transitive relationships.
- **Categories.** An entity can be associated with categories, which describe some aspect of its semantics, e.g., “*Big four consultants*” or “*XIX century composers*”. A book can belong simultaneously to all these categories: “*Books about Africa*”, “*Bestseller*”, “*Books by Italian authors*”, “*Books for kids*”, etc. The categories are described and ordered into taxonomy.
- **Free text descriptions.** Often a ‘human-friendly text’ description is provided to further clarify design intentions for the entity and improve search.

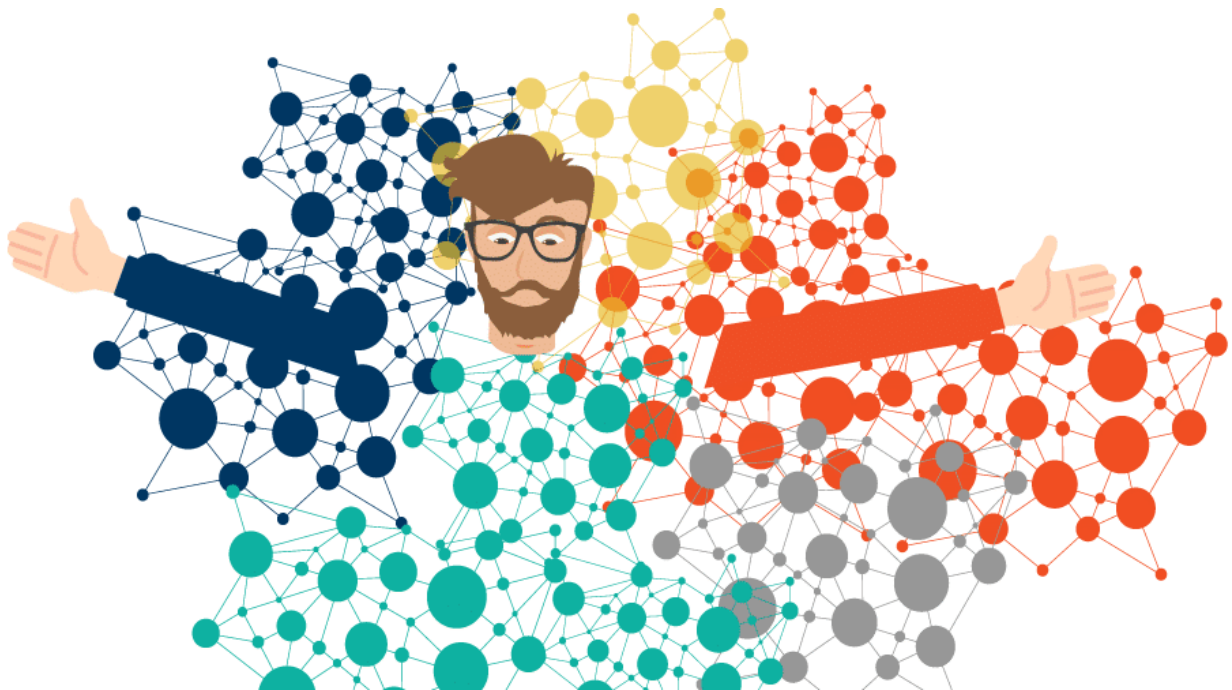
What is NOT a Knowledge Graph?

Not every RDF graph is a knowledge graph. For instance, a set of statistical data, e.g. the GDP data for countries, represented in RDF is not a KG. A graph representation of data is often useful, but it might be unnecessary to capture the semantic knowledge of the data. It might be sufficient for an application to just have a string ‘Italy’ associated with the string ‘GDP’ and a number ‘1.95 trillion’ without needing to define what countries are or what the ‘Gross Domestic Product’ of a country is. It’s the connections and the graph that make the KG, not the language used to represent the data.

Not every knowledge base is a knowledge graph. A key feature of a KG is that entity descriptions should be interlinked to one another. The definition of one entity includes another entity. This linking is how the graph forms. (e.g. A is B. B is C. C has D. A has D). Knowledge bases without formal structure and semantics, e.g. Q&A “knowledge base” about a software product, also do not represent a KG. It is possible to have an expert system that has a collection of data organized in a format that is not a graph but uses automated deductive processes such as a set of ‘if-then’ rules to facilitate analysis.

Examples of Big Knowledge Graphs

Google Knowledge Graph. Google made this term popular with the announcement of its knowledge graph in 2012. However, there are very few technical details about its organization, coverage and size. There are also very limited means for using this knowledge graph outside Google's own projects.



DBpedia. This project leverages the structure inherent in the infoboxes of Wikipedia to create an enormous dataset of 4.58 things (link <https://wiki.dbpedia.org/about>) and an ontology that has encyclopedic coverage of entities such as people, places, films, books, organizations, species, diseases, etc. This dataset is at the heart of the Open Linked Data movement. It has been invaluable for organizations to bootstrap their internal knowledge graphs with millions of crowdsourced entities.

Geonames. Under a creative commons, users of Geonames dataset have access to 25 million geographical entities and features.

Wordnet. One of the most well-known lexical databases for the English language, providing



definitions and synonyms. Often used to enhance the performance of NLP and search applications.

FactForge. After years of developing expertise in the news publishing industry, Ontotext produced their knowledge graph of Linked Open Data and news articles about people, organizations and locations. It incorporates the data from the KGs described above as well as specialized ontologies such as the Financial Industry Business Ontology.

Knowledge Graphs and RDF Databases

Years ago, we moved away from the buzzword of Big Data to Smart Data. Having unprecedented amounts of data pushed the need to have a data model that mirrored our own complex understanding of information.

To make data smart, the machines needed to be no longer bound by inflexible data schemas defined 'a priori'. We needed data repositories that could represent the 'real world' and the tangled relationships that are entailed. All this needed to be done in a machine-readable way and have a formal semantics to enable automated reasoning that complemented and facilitated our own.

RDF databases (also called RDF triplestores), such as Ontotext's [GraphDB](#), can smoothly integrate heterogeneous data from multiple sources and store hundreds of billions of facts about any conceivable concept. The RDF graph structure is very robust (it can handle massive amounts of data of all kinds and from various sources) and flexible (it does not need its schema re-defined every time we add new data).



As we have already seen, there are many freely available interlinked facts from sources such as DBpedia, GeoNames, Wikidata and so on, and their number continues to grow every day. However, the real power of knowledge graphs comes when we transform our own data into [RDF triples](#) and then connect our proprietary knowledge to open global knowledge.

Another important feature of RDF databases is their [inference](#) capability where new knowledge can be created from already existing facts. When such new facts are materialized and stored in an RDF database, our search results become much more relevant, opening new avenues for actionable insights.

But if we want to add even more power to our data, we can use text mining techniques to extract the important facts from free-flowing texts and then add them to the facts in our database.

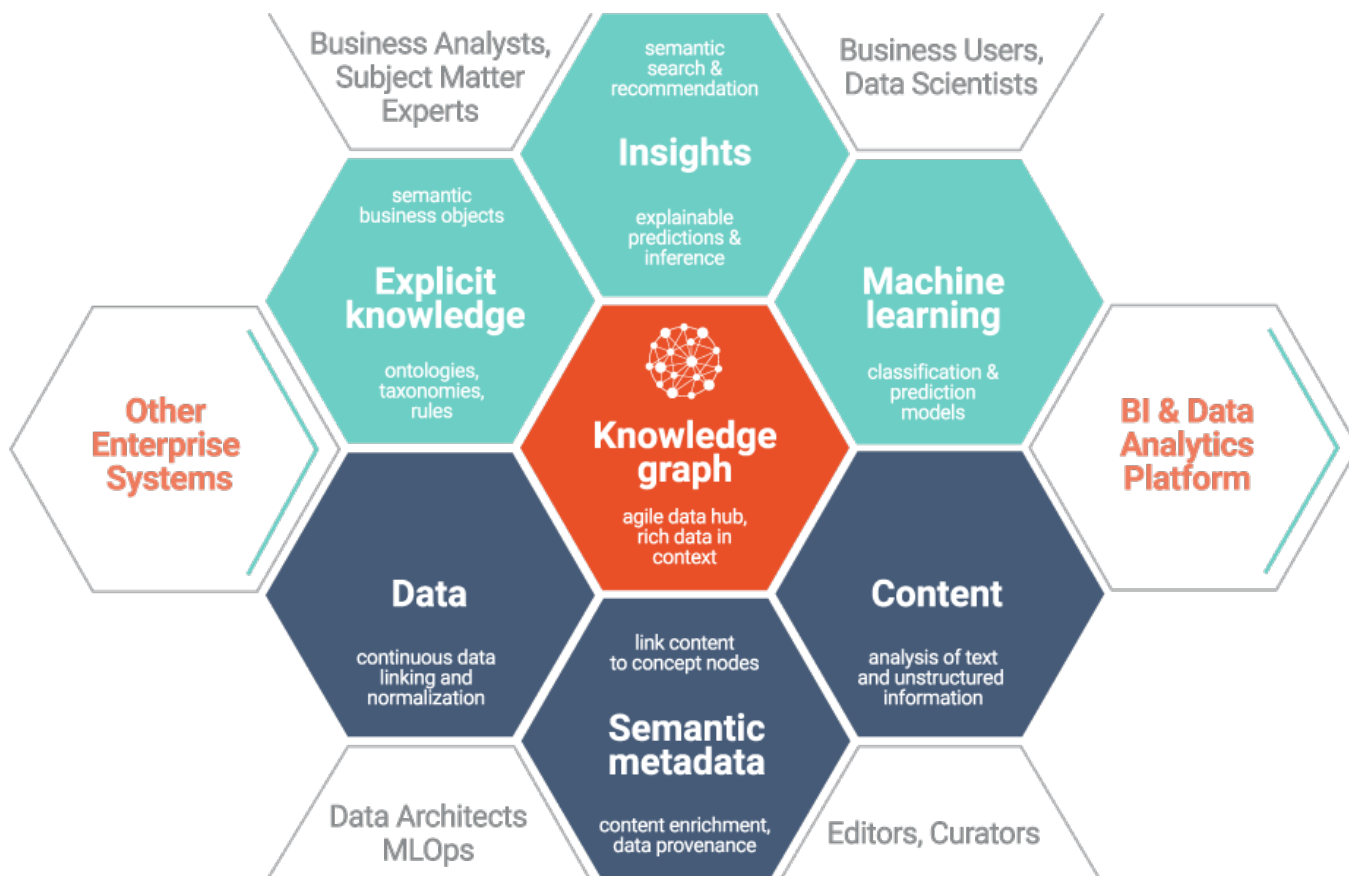
How Can Knowledge Graphs Help Text Analysis

It is no surprise that modern [text analysis](#) technology makes considerable use of knowledge graphs:



- Big graphs provide background knowledge, human-like concept and entity awareness, to enable a more accurate interpretation of the text;
- The results of the analysis are semantic tags (annotations) that link references in the text to specific concepts in the graph. These tags represent structured [metadata](#) that enables better search and further analytics;
- Facts extracted from the text can be added to enrich the knowledge graph, which makes it is much more valuable for analysis, visualization and reporting.

[Ontotext Platform](#) implements all flavors of this interplay linking text and big knowledge graphs to enable solutions for [content tagging](#), [classification](#) and [recommendation](#). It is a platform for organizing enterprise knowledge into knowledge graphs, which consists of a set of databases, machine learning algorithms, APIs and tools for building various solutions for specific enterprise needs.





One interesting example of semantic tagging on news against a big knowledge graph developed around DBpedia is Ontotext's [NOW public news service](#).

What Are Knowledge Graphs Used for?

A number of specific uses and applications rely on knowledge graphs. Examples include data and information-heavy services such as [intelligent content and package reuse](#), [responsive and contextually aware content recommendation](#), [knowledge graph powered drug discovery](#), [semantic search](#), [investment market intelligence](#), [information discovery in regulatory documents](#), [advanced drug safety analytics](#), etc.

Do you want to try building your knowledge graph?

Download GraphDB and start building knowledge graphs for your data management practices!

Give It a Try