Cite as: Bergman, M. K. Knowledge Graphs and Bases. in *A Knowledge Representation Practionary: Guidelines Based on Charles Sanders Peirce* (ed. Bergman, M. K.) 227–247 (Springer International Publishing, 2018). doi:10.1007/978-3-319-98092-8_11

Official site: https://link.springer.com/book/10.1007/978-3-319-98092-8

Full-text: http://www.mkbergman.com/publications/akrp/chapter-11.pdf

Abstract: Relations between nodes, different than those of a hierarchical or subsumptive nature, provide still different structural connections across the knowledge graph. Besides graph theory, the field draws on methods including statistical mechanics from physics, data mining and information visualization from computer science, inferential modeling from statistics, and social structure from sociology. We want knowledge sources, putatively knowledge bases, to contribute the actual instance data to populate our ontology graph structures.

11

KNOWLEDGE GRAPHS AND BASES

Virtually everywhere one looks we are in the midst of a transition for how we organize and manage information, indeed even relationships. Social networks and online communities are changing how we live and interact. NoSQL and graph databases — married to their near cousin 'big data' — are changing how we organize and store information and data. Semantic technologies, backed by their ontologies and RDF data model, are showing the way for how we can connect and interoperate disparate information in ways only dreamed about a decade ago. Moreover, we are building all of this upon the infrastructure of the Internet and the Web, a global, distributed network of devices and information that is undoubtedly one of the most significant technological developments in human history.

The graph is a shared structure across all of these developments.[1] Graphs are the new universal paradigm for how we organize and manage information. Graphs have an inherently expandable nature and one which can also capture any existing structure. So, as we see all of the networks, connections, relationships, and links — both physical and informational — grow around us, it is useful to step back a bit and contemplate the universal graph structure at the core of these developments. Some form of conceptual schema governs every knowledge structure used for knowledge representation (KR) or knowledge-based artificial intelligence (KBAI). In the semantic Web space, we call such schema 'ontologies.' Because the word ontology is a bit intimidating, a better variant is the *knowledge graph* (because all semantic ontologies take the structural form of a graph). In our knowledge representation efforts, we tend to use the terms *ontology* and *knowledge graph* interchangeably.

What an ontology — or knowledge graph — means is dependent on context and purpose. In the case of an *upper ontology* and *typologies*, we see the conceptual scaffolding. In the relation of *attributes* to *instances*, we see the intensional aspects of the graph and the basis for data records. Relations between nodes, different than those of a hierarchical or subsumptive nature, provide still different structural connections across the knowledge graph. Indeed, one can and should organize the types of *types* in a knowledge graph to better modularize it and segregate similar purposes and functions. We design some ontologies to capture the scope of particular knowledge domains, while others we may use for administrative purposes or in support of user interfaces. We discuss all of these aspects in this chapter, plus what is desirable in knowledge bases and how to use them to populate these knowledge structures.

GRAPHS AND CONNECTIVITY

Graphs, as conceptual or analytical structures, are relatively new. The explication of graph theory only began about 300 years ago. The use of graphs for expressing logic structures only began about 100 years ago, with its intellectual roots, in fact, arising from Charles Peirce and his existential graphs. Though likely trade routes and primitive transportation or nomadic infrastructures were perhaps the first expressions of physical networks, the emergence and then prevalence of networks is also a fairly recent phenomenon. Transportation, communications, and the electrical grid were the first purpose-built physical networks. The Internet and the Web are surely the catalyzing development that has brought graphs and networks to the forefront.

In mathematics, a graph is an abstract representation of a set of objects where pairs of the objects are connected. We term these objects *nodes* or *vertices*; we call the connections between the objects *edges*. Typically, we depict a graph in diagrammatic form as a set of dots or bubbles for the nodes, joined by lines or curves for the edges. If we define a logical relationship between connected nodes we call the graph 'directed.' We can express various structures or topologies through this conceptual graph framework. Graphs are one of the focuses of study in discrete mathematics.[2] The word 'graph' was first used in a mathematical sense by J.J. Sylvester in 1878.[3]

Graphs are modular and can be both readily combined and broken apart. From a computational standpoint, this can lend itself to parallelized information processing (and, therefore, scalability). If we represent the graph in RDF, graph extractions are themselves valid models. Graphs have some unique strengths for search and pattern matching. Besides options like finding paths between two nodes, depth-first search, breadth-first search, or finding shortest paths, emerging graph and pattern-matching approaches may offer entirely new paradigms for search. Graphs also provide new methods for visualization and navigation, useful for both seeing relationships and framing information from the local to global contexts. The interconnectedness of the graph allows us to explore data via contextual facets, which is revolutionizing data understanding in a way similar to how the basic hyperlink between documents on the Web changed the contours of our information spaces.

Graph algorithms are a significant field of interest within mathematics, computer science, and the social sciences. Via approaches such as network theory or scale-free networks, we can analyze and model topics such as relatedness, centrality, importance, influence, 'hubs' and 'domains,' link analysis, spread, diffusion and other dynamics. Many would argue, as do I, that graphs are the most 'natural' data structure for capturing the relationships of the real world. If so, we should continue to see new algorithms and approaches emerge based on graphs to help us better understand our information. RDF is a natural data model for such purposes.

Graph Theory

Graph theory is the manipulation and analysis of graph structures. The first paper in that field is the *Seven Bridges of Königsberg*, written by Leonhard Euler in 1736. The objective of the article was to find a walking path through the city that would cross each bridge once and only once. Euler proved that the problem had no solution.* Later, Cayley broadened the approach to study tree structures, which have many implications in theoretical chemistry. By the 20th century, the fusion of ideas coming from mathematics with those coming from chemistry formed the origin of much of the standard terminology of graph theory.

Graph theory forms the core of network science, the applied study of graph structures and networks. Besides graph theory, the field draws on methods including statistical mechanics from physics, data mining and information visualization from computer science, inferential modeling from statistics, and social structure from sociology. Classical problems embraced by this realm include the four color problem of maps, the traveling salesman problem, and the six degrees of Kevin Bacon. Graph theory and network science are the suitable disciplines for a variety of information structures and many additional classes of problems. Graphs are among the most ubiquitous models of both natural and human-made structures. They can be used to model many types of relations and process dynamics in physical, biological and social systems. Graphs can represent many problems of practical interest. This breadth of applicability makes network science and graph theory two of the most critical analytical areas for study and breakthroughs for the foreseeable future.

Graphs and graph theory also have broad applicability to natural systems. For instance, researchers use graph theory extensively to study molecular structures in chemistry and physics. A graph makes a natural model for a molecule, where vertices represent atoms and edges bonds. Similarly, in biology or ecology, researchers employ graphs to express such systems as species networks, ecological relationships, migration paths, or the spread of diseases. Graphs are also proper structures for modeling biological and chemical pathways. Some of the exemplary natural systems that lend themselves to graph structures include:

- Chemical reaction networks
- Gene regulatory networks
- Spin networks
- Neural networks
- Ecological networks, and
- Petri nets (chemistry).

The growth of social networks has paralleled the growth of the Internet and Web. Social network analysis (SNA) has arguably been the most critical driver for advances in graph theory and analysis algorithms in recent years. We are now elucidating new

* The generalized understanding is that in any connected graph, only zero or two nodes may have odd numbers of connections to traverse the entire graph only once per path (edge); the Königsberg example has four nodes with odd numbers, and thus fails Euler's test.

and interesting problems and challenges — from influence to communities to conflicts — through techniques pioneered for SNA. The suitability of the graph structure to capture relationships has been a real boon to a better understanding of social and community dynamics. SNA has introduced many new concepts, including such things as influence, diversity, centrality, and cliques. Particular areas of social interaction that lend themselves to SNA include:

- Social networks
- Military conflicts and terrorism
- Value networks
- Project networks
- Workflows, and
- Business ecosystems.

We have unearthed entirely new insights using SNA including finding terrorist leaders, analyzing prestige, or identifying keystone vendors or suppliers in business ecosystems. Real networks, in comparison to random networks, are both modular and hierarchical, distributed over a sparse topology.[14]

What these examples show is the nearly universal applicability of graphs, from the abstract to the physical and gradations from the small to the large. We also see how to build upon basic graph structures and concepts with more structure. This breadth points to the many synergies and innovations that may be transferred from diverse fields to advance the usefulness of graph theories. Still, despite the advances that have occurred in graph theory, and the increased attention from social network analysis, many graph problems remain some of the hardest in computation. Optimizations, partitioning, mapping, inferencing, traversing and graph structure comparisons remain challenging. Some of these challenges are only growing due to the growth in the size of networks and graphs.

Given the ubiquity of graphs as representations of real systems and networks, it is not surprising to see their use in computer science as means for information representation. It is notable that we may represent virtually any data structure as a graph, but the paradigm has even broader applicability. The critical breakthroughs have come through using the graph as a basis for data models and logic models. These, in turn, provide the basis for crafting entire graph-based vocabularies and languages. Once we embrace such structures, it is also natural to extend the mindset to graph databases as well.

The Value of Connecting Information

The hackneyed phrase of ‘connect the dots’ reflects our basic intuition of the value in making connections amongst relevant data. However, what is this value? How might we quantify it? The reason it is useful to try to quantify the value of connected information is that such an estimate helps to define what effort or cost we can justify building our connected knowledge structures. For most big data projects, for example, we already know that 50% to 80% of the costs in assembling relevant datasets is due to data wrangling — the effort to extract, transform and clean the input data.[5] Nowhere, however, do we know what it is worth to go to the next step of working to connect those data.

The ‘network effect‘ was first realized in the early days of telephone networks, where the value of the system increased as a function of more users.[6] We have also long recognized a similar effect in connecting information and the breaking down of information or ‘data silos.‘ This emergence of structure is particularly evident in physical networks, such as the growth of a telecommunications network. Two telephones can make only one connection, five can make ten connections, and twelve can make 66 connections, etc. It is this very multiplier effect that has led to most of the thinking of how to quantify the network effect.

The earliest effort to estimate the value of physical networks was Sarnoff’s law, developed by David Sarnoff, for many years the leader of the Radio Corporation of America (RCA). He posited that the value of a broadcast network was directly proportional to its number of viewers (*n*). However, the problem with this formulation is that a broadcast network is only one way, from broadcaster to user. What of networks with interactions or two-way linkages? The benefits of such networks must surely be more than linear.

Once we get into interaction effects, we get into multipliers. The nature of those multipliers come from the extent of real interactions, as well as perhaps the nature of the network itself. Metcalfe’s law was the first direct derivation from the telecommunications model. Robert Metcalfe formulated it about 1980 in relation to Ethernet and fax machines. The ‘law’ was then named for Metcalfe and popularized by George Gilder in 1993.[7] The actual algorithm proposed by Metcalfe calculated the number of unique connections in a network with n nodes as $n(n - 1)/2$. This formulation makes Metcalfe’s law a quadratic growth equation. We may simplify the law[8] to state that the value of a telecommunications network is proportional to the square of the number of users of the system (n^2). Gilder’s popularization and the early growth of the Internet made estimating the benefits of network effects a very timely topic. As a value measure, we can use the network effect to estimate the benefits for increasing numbers of users. Some have even blamed Metcalfe’s law for contributing to the creation (and then bursting) of the ‘dot-com bubble’ of the late 1990s.[9]

However, the Metcalfe formulation is not universally accepted, and others have proposed different estimates. From the perspective of social groups, Reed came up with the largest multiplier formulation premised on arbitrary sized groups forming amongst any and all participants (nodes).[10] On the other hand, under the provocative title, “Metcalfe’s Law is Wrong,” Briscoe, Odlyzko, and Tilly (BOT) challenged both the Metcalfe and Reed approaches in 2006.[11] Using the proxy of Internet valuation, the authors were able to show how absurd the implications of either approach were at scale. Like the bet of rice (or wheat) doubling each of the 64 squares on a chessboard bankrupting the kingdom, we can see the exponential implications of these two ‘laws’ to (eventually) violate common sense. The fundamental fallacy claimed by the authors for both the Metcalfe and Reed approaches is that all potential links are of equal value. There must be some law of diminishing returns to slow the unsustainable rates of exponential or (to a lesser extent) quadratic growth. After much hand waving, the authors chose Zipf's law[12] as their basis for this diminishing return. To approximate this distribution, they (BOT) offered the simple *n log (n)* formulation of Zipf's law. This approximation is reasonable, but one that is never related directly to the real nature of graphs or networks.

Yaakov Stein, a network and signals processing researcher of the first rank, used his experience when joining LinkedIn to help understand and quantify connections in real networks. [13] He began without a LinkedIn account and documented his experience as he joined and expanded his network of contacts on the service. He charted direct links, and then meticulously looked at and recorded secondary and tertiary links. His formulation recognized that the value to an individual user equaled raising the access to the entire network (*1*) for that user plus the diminishing benefit represented by the participating graph's other participants as measured by the average degree of separation (*D*). *D* is an inherent measure of the graph type.

Though his context was a social network, the insight is that relations diminish by distance within a graph, with average link distance (directly related to the degree of separation) a relevant metric. Connected 'facts' or 'friends' is essentially the same thing. It is all about what we share amongst graph nodes. Stein's approach grounds the multiplier effect in an inherent characteristic of the graph: its average degree of separation. Like Zipf's law, the degree of separation is a distance measure, but one now based on the real nature of graphs. Here is the Stein formulation:

$$V = n^{(1+1/D)}$$

where *V* is potential value, *n* is number of graph nodes, and *D* is the graph's average degree of separation. Thus, a graph with a degree of separation of 4, would exhibit a network-wide power factor of 5/4 (4/4 plus 1/4).

I modified Stein's approach to calculate the Value of Knowledge Graph formulation, or the VKG (Viking) algorithm, using this expression:

$$V = F * n^{(1+1/D)}$$

where *V* is potential value, *F* is average assertion accuracy, *n* is number of graph nodes, and *D* is the graph's average degree of separation. *F* is analogous to F-measure, the combined precision and recall statistic for information retrieval and NLP tasks (see further *Chapter 14*). *F* in the case of the Viking algorithm is also a combined statistic that represents the 'accuracy' (verifiable truthfulness) of statements asserted in the graph.

F is essentially an estimated value for one minus the residual falsity for the average statement in a graph, after removal of all assertions that do not meet existing coherency, consistency or completeness tests. Sampling statements across the graph determine *F* and manually testing for truthfulness (or in a logical sense, validity for the existing statements in the graph). An *F* of 1 signifies complete truthfulness (accuracy); an *F* of 0 represents absolute falsity.

Now corrected with our assumed *F* factor, we can begin to tease out the value benefits of connecting 'facts' versus the unconnected 'facts.' As with any logarithmic function, we see that the benefit value from connections increases in a growing manner at larger scales. For example, at a level of 1000 records, the benefits from connections are 7x greater than unconnected data. By the time the scale grows to 1 million or 500 million records, the benefit of connections increases to 44x to 215x, respectively. However, the potential value of connectedness is also a function of the general degree of information separation for the given domain.

I consolidate these various estimates of connected value in *Figure 11-1*. At the nominal scales of 100,000 and 1,000,000 records, the value of data connections in comparison to the unconnected 'facts' case can show huge increases. Based on empirical experience to date, I think we can say the benefits of connecting previously unconnected data may fall somewhere within the limits of *Figure 11-1*. Even at rather low scales and more loosely-connected domains, the value improvements in making connections with data are many-fold. At larger scales for tighter networks, the multipliers can become astounding.

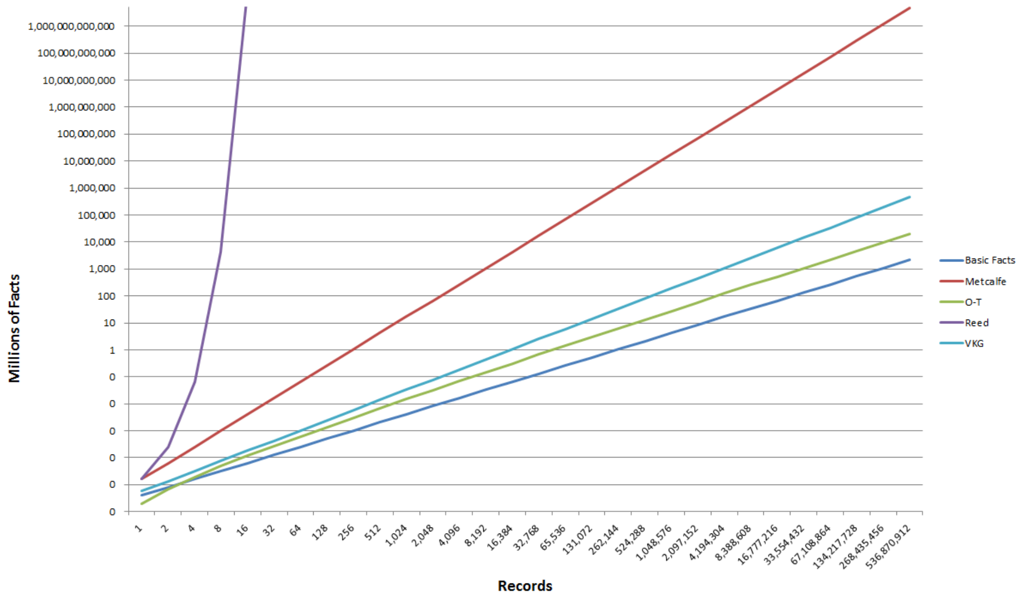

Figure 11-1: Comparison of Network Multipliers

We are still in the early phases of gathering statistics for such things, but, in general, most any knowledge graph would have a *D* factor ranging from 2 to 8, as I document in *Table 11-1*. We also should assume network effects are not linear. We should expect a leveling or flattening of the curve; the benefits of connections are not limitless. The shape of the curve likely varies by domain and the nature of the network. It is a topic worth studying.

Domain	*100,000 Records*	*1,000,000 Records*
Food webs	203x	611x
Genetic differences	38x	84x
Twitter	23x	46x
Facebook	17x	33x
Potential research collaborators	14x	26x
UMBEL	8x	12x
Social networks (general)	5x	8x
Mobile ad hoc networks	3x	5x

Table 11-1: Increased Value for Connecting Nodes, Various Networks

Another implication the Viking algorithm allows us to test is the benefit of adding structure to our datasets. Actually, 'adding structure' is not strictly correct; it is 'structurizing' the data via characterizations, attributes and categorizations. Of course, mere connections for structure's sake is silly. Not all structure is created equal. From a KR perspective, typing is the most important, individual instance annotations the least. Assigning or classifying our records into types, for example, applies to all records across the datasets and provides powerful cross-record linkages. Adding annotations or metadata to single records provides much lower benefits. Each across-dataset structure characterization adds about 25% to 30% value per structure. Adding four structural characterizations, for example, more than doubles the 'facts' assertion value (~ 140%) to the datasets. The good thing is that we can add such structure as a slight increase over standard data wrangling efforts, and with more impact than standard wrangling.

The graph structures, preferably guided by domain ontologies, provide the logic means to test for subsequent structure additions. Not only does adding structure get easier with a foundation of existing structure, but it increases the value of the information by orders of magnitude. At this stage, what the Viking algorithm gives us is a defensible means for assessing the value of adding structure (through connections) to our datasets. We see potentially huge multiplier effects that compound further benefits with scale. (Subject to the leveling curve caveat.) We also see that the most developed forms of structure — namely, ontologies — bring further benefits in inference and testable coherence.[14]

While our current proxy for value — namely, asserted 'facts' — is useful, a perhaps more useful one would be 'fact' assertions with a monetary value. Such estimates will show, again, that not all 'facts' are created equal, and some have more monetary value than others. Transitioning our estimates of value to a monetary basis will help set parameters for the cost-benefit analysis of data collection and structuring that is the ultimate basis for planning such KR initiatives.

As we look at *Table 11-1* and play with some parameters, we can see some guidelines emerge. First, more structure always provides benefits — adding structure provides a multiplier effect in value. Second, more connections are more valuable as a multiplier effect than adding more data, which has an additive effect. Third, the benefits of structure increase with increasing dataset sizes (scale). Fourth, particular kinds of structure, such as types or categorizations, enable cross-dataset selections and comparisons that are inherently more valuable than record-specific annotations. Fifth, by adding correct and coherent connections, it may be possible to move the entire graph to a lower average degree of separation (*D*), with further multiplier benefits. Sixth, structure can be added incrementally and appears cumulative to some level. Seventh, we should not view data wrangling as an overall 'cost' to the effort but as a means for achieving the multiplier benefits arising from structure and connections.

Graphs as Knowledge Representations

Graphs are an iconic and intuitive way to visualize and express connections. Graphs, expressed in mathematical or logical form, are a rich substrate for analysis and reasoning. Graphs appear to be the natural structure for capturing real relationships in the material world and the conceptual realm.

One key aspect of graphs is their inherent extensibility. Once we understand graphs as an excellent way to represent both logic and data structures, their usefulness to knowledge representations becomes clear. Graph-theoretic methods are particularly useful in linguistics since natural language has a discrete, connected structure. Not only can graphs represent the syntactic and compositional structure, but they can also capture the interrelationships of terms and concepts within those languages (that is, the *semantics*). We see the usefulness of graph theory to linguistics by the various knowledge bases such as WordNet (in multiple languages) and VerbNet. Domain ontologies emphasize conceptual relationships over lexicographic ones for a given knowledge domain. Semantic networks and neural networks are similar knowledge representations.

The main reasoning in the knowledge graph relies on its hierarchical, hyponymous relations and instance types. These establish the parent-child lineages and enable us to relate individuals (or instances, which might be entities or events) to their natural kinds, or types. Entities belong to types that share specific defining essences and shared descriptive attributes. For effective inferencing, it is wise to try to classify entities into the most natural kinds possible. Clean classing into appropriate types is one way to realize the benefits from related search and related querying. Types may also have parental types in a hyponymous relation. This 'accordion-like' design, discussed in the prior *Chapter 10*, is an important aspect that enables us to tie external schema to multiple points in KBpedia.

Disjointedness assertions, where two classes are logically distinct, and other relatedness options provide other powerful bases for winnowing potential candidates in a graph and testing placements and assignments. Each of these factors also may be used in SPARQL queries. These constructs of semantic Web standards, combined with a properly constructed knowledge graph and the use of synonymous and related vocabularies in *semsets*, provide potent mechanisms for how to query a knowledge base. By using these techniques, we may dial-in or broaden our queries, much in the same way that we choose different types of sprays for our garden watering hose. We can focus our queries to the particular need at hand.

Once a completed graph passes its logic tests during construction, perhaps importantly after being expanded for the given domain coverage, its principal use is as a read-only knowledge structure for making subset selections or querying. The standard SPARQL query language, which we occasionally supplement with rule-based queries using SWRL or for bulk actions using the OWL API, is the means by which we access the knowledge graph in real time. In many instances, such as for the KBpedia knowledge graph, these are patterned queries. In such cases, we substitute variables in the queries and pass those from the HTML to query templates. When doing machine learning, we often retrieve slices via query and then stage them for the learner. We may generate entity lists for things like training recognizers and taggers. Some of the actions may also do graph traversals to retrieve the appropriate subset. However, the primary real-time use of the knowledge structure is search.

Among many other options, SPARQL also gives us the ability to query specific property paths.[15] We can invoke these options either in our query templates or programmatically. We may programmatically broaden or narrow our searches of the graph, depending on the relation chosen (`subClassOf` is one example) and the length of the specified property path. Switching inferencing on or off also acts to broaden or narrow the search swath considerably. Besides all of the standard query options provided by the SPARQL standard, we may also remove duplicates, identify negated items, and search inverses, select named graphs, or select graph patterns. Beyond SPARQL and now using SWRL, we may also apply abductive reasoning and hypothesis generation to our graphs, as well as mimic the action of expert systems in AI through if-then rule constructs based on any structure within the knowledge graph. A helpful online tutorial with examples helps highlight some of the possibilities in combining OWL 2 with SWRL.[16]

UPPER, DOMAIN AND ADMINISTRATIVE ONTOLOGIES

The root of the *ontology* term is the Greek *ontos*, or *being* or the *nature of things*. Classical philosophers used the term ontology for the study of the nature of being or the world, the nature of existence. Tom Gruber, among others, made the term popular in computer science and artificial intelligence about 15 years ago when he defined ontology as a "formal specification of a conceptualization." Since then, I have continued to find ontology one of the harder concepts to communicate to clients and quite a muddled mess even as used by some practitioners. I have concluded that this problem is not because I have failed to grasp some ephemeral nuance, but because the 'ontology' term as used in practice is indeed fuzzy and imprecise.

A Lay Introduction to Ontologies

Ontologies are the structural frameworks for organizing information on the semantic Web and within semantic organizations.* Ontologies have the structural form of a graph; we often use *knowledge graph* synonymously. Ontologies provide unique benefits in discovery, flexible access, and information integration due to their inherent connectedness. We can layer ontologies on top of existing information assets, which means they are an enhancement and not a displacement for prior investments. Moreover, ontologies may be developed and matured incrementally, which means their adoption may be cost-effective as benefits become evident.

Ontologies provide an organizing context for relating disparate information together and for making meaningful inferences. The framework itself is a function of the worldview, context and domain scope at hand. Flexibility here is not weakness; it is the power to capture the meaningful vocabulary and discourse for entire domains of knowledge. The trick to designing a proper ontology is to maintain internal coherence and self-consistency while capturing the vocabulary and discourse of its stakeholders and users. When done, it is then possible to relate disparate information and data to other data and to make intelligent business inferences. So, the use of an ontology does not limit freedom. It does set the context for making connections and setting relations. As long as it is coherent, the 'correct' ontology is the one that best captures the scope and domain at hand, and is one that is continually responding to the open nature of knowledge and its community of users.

When I refer to the idea of 'worldview' as synonymous with an ontology, I do not mean that as cosmic, but how we may convey a given domain or problem area. One group might choose to describe and organize, say, automobiles, by color; another might choose body styles such as pick-ups or sedans; or, still, another might use brands such as Honda and Ford. None of these views is inherently 'right' (indeed multiples might be combined in a given ontology), but each represents a particular way — a 'worldview' — of looking at the domain. So long as all ascertainable 'facts' in an ontology may be confirmed and its logic kept consistent, different 'worldviews' are perfectly acceptable.

Understanding, using and manipulating ontologies can bring practical benefits:

- Ontologies help make explicit the scope, definition, language, and meaning (*semantics*) of a given knowledge domain or worldview;
- Ontologies may represent any form of *unstructured* (documents or text), *semi-structured* (XML or Web pages) or *structured* (database) data;
- Ontologies provide a coherent navigation and search mechanism for moving through disparate information spaces, with any *node* or *edge* providing a possible entry point;
- Ontologies, if hierarchically structured in part, enable the power of inheritance, reasoning, and inference;

* I personally prefer an embracing understanding of the term, consistent with Deborah McGuinness's 2003 paper, *Ontologies Come of* Age.[17]

- Ontologies may provide the power to generalize and hypothesize (*abductive reasoning*) about their domains;
- Ontologies provide guidance on how to correctly 'place' information in that domain, useful for external concept matching and mapping;
- Ontologies can provide a more effective basis for information extraction or content clustering ;
- Ontologies may be queried and filtered to provide pre-qualified corpora and training sets, useful to *unsupervised* and *supervised machine learning*, respectively;
- Ontologies may be a source of structure and controlled vocabularies helpful to disambiguate context and to inform domain 'lexicons';
- Ontologies can help relate and 'place' other ontologies or worldviews to one another; in other words, ontologies can help organize ontologies.

The most prevalent use of ontologies at present is in semantic search. Semantic search has benefits over conventional search by being able to make inferences and matches not available to standard keyword retrieval. Perhaps a pinnacle application for ontologies is to help map and integrate other structures and information, both within and without the organization. Furthermore, if we populate a knowledge graph sufficiently with accurate instance data, often from various knowledge bases, then ontologies can also be the guiding structures for efficient machine learning and artificial intelligence.

Ontologies are A Family of Graphs

If you pose the query 'ontology filetype:owl' to Google, you will see more than 10,000 results. According to Ontolog Forum, a community of ontology practitioners, we can classify ontologies by some key measures. *Expressiveness* is the extent and ease by which an ontology can describe domain semantics. *Structure* they define as the degree of organization or hierarchical extent of the ontology. They further define *granularity* as the level of detail in the ontology. By these notions, we may include the concepts of *folksonomies* and *topic maps* in the definition. The Forum also defines other dimensions of use, logical basis, and purposes for ontologies.[18] One of these dimensions is to characterize ontologies by 'levels,' specifically *upper*, *middle* and *lower* levels. These are useful distinctions, but we prefer to classify them into *upper*, *domain* and *administrative* ontologies.

Upper ontologies provide the top-level conceptual structure and schema, which often function as the reference structure for specific domain ontologies. Examples of upper-level ontologies include the Suggested Upper Merged Ontology (SUMO), the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE), PROTON, Cyc and BFO (Basic Formal Ontology). Most of the content in these upper-levels is akin to broad, abstract relations or concepts. Most all of them have both a hierarchical and networked structure, though their actual subject structure relating to concrete things is pretty weak. KBpedia's Knowledge Ontology (KKO) is an example of an upper ontology.

Domain (or content) *ontologies* embody more of the traditional ontology functions such as information interoperability, inferencing, reasoning and conceptual and knowledge capture of the applicable domain. We can broadly or narrowly define these domains; specific instantiations may cover multiple or a diverse range of subject matter. In KBpedia’s design, we compose the domain ontology of multiple typologies, including for relations and attributes.

Administrative ontologies govern internal application use and user interfaces. These areas might relate to providing metadata as a result of workflow steps and general workflow management, as well as driving visualization or display widgets or informing user interfaces. Possible user interface aids provided by administrative ontologies may include attribute labels and tooltips; navigation and browsing structures and trees; menu structures; auto-completion of entered data; contextual dropdown list choices; spell checkers; and online help systems. Administrative ontologies may also support internal applications such as workflow systems, access control, archive management, and the like.

Incipient Potentials

For over twenty years, some researchers such as Nicola Guarino (1998)[19] and Michael Uschold (2008)[20] have argued that we could rely upon ontologies for even more central aspects of overall applications, what Uschold termed ’ontology-driven information systems.’ I agree. Here are five areas of (largely) untapped potential:

1. Lack of a well-known *relations ontology*. Structurally, we may use OWL to reason over actions and relations in a similar means as we reason over entities and types, but our common ontologies have yet to do so. Creating such schema is within grasp since we have language structures such as VerbNet and other resources we could put to the task. KBpedia has its own relations typologies that attempt to capture these aspects;
2. Lack of a well-known *attributes ontology*. The lack of a schema and organized presentation of attributes means it is challenging to do *ABox*-level integration and interoperability. This gap is largely due to the primary focus on concepts and entities in the early stages of semantic technologies. As the KBpedia knowledge graph shows, it is possible to formulate logical and reusable schema for instance attributes as well;
3. A *quantity units ontology* is the next step beyond attributes, as we attempt to bring data values for quantities (and well as the units and labeling used) into alignment. The QUDT ontologies (quantities, units and data types), or something similar, may provide such a template;
4. A *statistics and probabilities ontology* is also appropriate given the idea of continua (Thirdness) from Peirce and capturing the idea of *fallibility*. Probabilistic reasoning is still a young field in ontology. Some early possibilities include

Costa[21] and the PR-OWL ontology using Multi-Entity Bayesian Networks,[22] probabilistic first-order logic that goes beyond Peirce's classic deterministic logic, as well as fuzzy logic applied to ontologies[23]; and

5. *ODapps* ('ontology-driven applications') are generic software packages driven by ontology specifications for specific applications. They may enable us to: 1) import or export datasets; 2) create, update, delete (CRUD) or otherwise manage data records; 3) search records with full-text and faceted search; 4) manage access control at the interacting levels of users, datasets, tools, and CRUD rights; 5) browse or view existing records or record sets, based on simple to possible complex selection or filtering criteria; or 6) process results sets through workflows of various natures, involving specialized analysis, information extraction or other functions.

Realizing these potentials will enable our knowledge management (KM) efforts to shift to the description, nature, and relationships of the information environment. Under this broadened understanding, we now give explicit focus to the actual concepts, terminology, and relations that comprise coherent ontologies, subject to the direct control and refinement by their users, the knowledge workers and subject matter experts.

Good Ontology Design and Construction

While *Chapter 14* focuses on best practices and includes a section on ontologies, it is worthwhile here to reiterate three design considerations that should go into the construction of an ontology. These three factors are *coherence*, *completeness*, and *scope*, introduced in prior chapters.

Coherence is a state of logical, consistent connections, a logical framework for intelligently integrating diverse elements. In the sense of a knowledge graph, this means we have drawn the right connections (edges or predicates) between the object nodes in the graph. Structure without coherence is where we have not drawn correct or complete connections. The nature of the content graph lacks logic. The hip bone is not connected to the thigh bone, but perhaps to something wrong or ludicrous, like the arm or cheekbone.

Completeness is to conform to some minimum standard of characterization. For KBpedia, we have set that minimum as a preferred label, robust set of alternative labels (*semset*), a definition, language characterization, and one or more types or parents. If we have information on attributes, we should include that as well. However, it is not necessary to discover and document all attributes, though we should add new ones as we encounter them. See further what we discussed for completeness for reference concepts (RCs) in *Chapter 9*.

S*cope* means we answer a series of questions in the positive for the ontology:

- Does the ontology provide balanced coverage of the subject domain?* This ques-

* The sense of 'balance' is from the perspective of the sponsor, roughly bounded by the topic domain at hand.

tion gets at the issue of properly scoping and bounding the subject coverage such that the breadth and depth are roughly equivalent;

- Does the ontology embed its domain coverage into a proper context? Re-using existing and well-accepted vocabularies and including concepts in the subject ontology that aid such connections is good practice;
- Are the relationships in the ontology coherent, per our earlier condition? and
- Has the ontology been constructed according to good practice?

If we can answer these questions affirmatively — including importantly the use of testing scripts throughout construction — then we deem the ontology ready for production-grade use.

The skills needed to create these ontologies are logic, coherent thinking, and domain knowledge. That is, any subject matter expert or knowledge worker likely has the skills required to contribute to useful ontology development and refinement. Ontology development is a trainable skill.

KBPEDIA'S KNOWLEDGE BASES

We want knowledge sources, putatively *knowledge bases*, to contribute the actual instance data to populate our ontology graph structures. Matching with knowledge bases can also point out gaps and oversights in our knowledge graphs that we should augment to provide better domain coverage. Sufficient instance data is an absolute essential if we are to use our knowledge structures for *supervised* or *unsupervised machine learning*, or what we call herein *KBAI*.

We want knowledge bases to define and populate attributes for their instances. This kind of information is what we see in a data record. The best knowledge bases have large data stores, all consistently characterized. We prefer large sources because we can spread the effort of mapping and conversion across more records.

We prefer knowledge bases that provide identity and information for disambiguation. Identity works in that we can point to authoritative references (with associated Web identifiers) for all of the individual things and properties in our relevant domain. We can use these identities to decide the 'canonical' form, which also gives us a common reference for referring to the same things across information sources or data sets. We also want richness in how to describe those things.

Besides our earlier criteria of consistency, coherence, and completeness, our desiderata for what we find useful in a knowledge base includes:

- *Comprehensive* — does the knowledge base support the domain scope at hand? Smaller, focused knowledge bases may be quite valuable if the overlap is good;
- *Referencable* — is the knowledge source authoritative? Does it use IRIs or URIs for referencing its objects?

Work is always required to bring the knowledge graph up to this level of coverage. This sense is different for a library, where 'balance' is from the perspective of the patrons.

- *Open Standards* — does it meet open standards? It is often easier to interoperate with open standards with more tooling available;
- *Computable* — does the KB support reasoning, inference, set selection, relations, attributes, data types, and retrieval? If so, incorporation is easier; and
- *Multi-lingual* — if not already multi-lingual, does it have a structure (such as ID *v* label-based) that supports multiple languages? Support for multiple languages increases usefulness and applicability.

The idea that we can purposefully craft knowledge bases to support knowledge-based artificial intelligence, or KBAI, flows from these kinds of realizations. We begin to see that we can tease out different aspects of a knowledge base, each with its logic and relation to the other aspects.

KBpedia KBs

As of 2018, about 20 different knowledge bases contribute the instance data and some key mappings to KBpedia. Six of these are primary ones, defined as both adding large numbers of instances but scope coverage to KBpedia as well. We have selected the secondary KBs based on their common usage or their ability to contribute more limited concepts and structure to the overall KBpedia.

Primary KBs

Wikipedia, the primary source for structure, concepts, and definitions, and Wikidata, the primary source for millions of instance data and a rich system of attributes, are the two most significant contributors to KBpedia. We use DBpedia as a source for direct machine-readable Wikipedia data. While we first root the conceptual schema of KBpedia in Peirce's universal categories, we use the OpenCyc and UMBEL knowledge bases to inform the construction of KBpedia's typologies. We extend KBpedia's geographical and geopolitical reach using the GeoNames knowledge base. Here is a bit longer description of each source, current as of mid-2018:

- *Wikipedia* is a crowdsourced, free-access and free-content knowledge base of human knowledge. It has more than 5 million articles in its English version. Nearly 35 million articles exist across all Wikipedias in about 280 different languages. Though not universal, most all recent AI advances leveraging knowledge bases have utilized Wikipedia in one way or another, due to its scope, quality, and open-access structure. Wikipedia is a common denominator in question answering and commercial natural language applications that leverage artificial intelligence, witness Siri, Watson, Cortana and Google Now, among others. Even Freebase, the core of Google's Knowledge Graph, did not blossom as a separate data crowdsourcing concern until its former owner, Metaweb, decided to bring Wikipedia into its system. More than 1000 research papers leverage Wikipedia for AI and NLP purposes,[24] Many other knowledge bases are derivatives or enhancements to Wikipedia in one way or another.* One is hard-pressed to identify any large-scale knowledge base, available in electronic form, that is being exploited as much for AI or semantic technology purposes;[26]

- *Wikidata* is a crowdsourced, open *knowledge base* of (currently) about 55 million structured *entity records*. Each record consists of *attributes* and values with robust cross-links to multiple languages. Wikidata is a crucial entities source;
- *Cyc* is a common-sense *knowledge base* developed over 20 years involving about 1000 person-years of effort. The smaller open-source OpenCyc version is the one we use in KBpedia; an OWL version was available until that project ended in 2017. A ResearchCyc version of the entire system is still available to researchers. The Cyc platform contains a dedicated logic language, CycL, and has many built-in functions in areas such as *natural language processing*, search, inferencing and the like. *UMBEL* is based on a subset of OpenCyc;
- *DBpedia* is a project that extracts structured content from *Wikipedia* and then makes that data available as *linked data*. Millions of entities are characterized by DBpedia in this way. As such, DBpedia is one of the largest — and most central — hubs for *linked data* on the Web;
- *GeoNames* integrates geographical data such as names of places in various languages, elevation, population, and others from multiple sources. We obtain nearly 800 feature descriptors from GeoNames for organizing geographic and geopolitical information, as well as millions of well-characterized and -defined place names and regions; and
- *UMBEL*, short for Upper Mapping and Binding Exchange Layer, is an *upper ontology* of about 35,000 reference concepts, designed to provide universal mapping points for relating different ontologies or schema to one another, and a vocabulary for aiding that mapping.

The combination of these sources, organized by Peirce's triadic universal categories and typologies in the KKO, makes KBpedia a singularly unique knowledge resource.

Secondary KBs

We have mapped about 15 leading external vocabularies and ontologies to KBpedia, with the first three playing a more prominent role. This listing of mappings is:

schema.org	This extendable vocabulary describes common things, businesses, and events on the Web. Major search engines, including Google, sponsor it. There are more than 700 types in the vocabulary. Millions of Web documents are marked up with this vocabulary.

* Though a bit dated, an 82-page technical report by Olena Medelyan *et al.* from the University of Waikato in New Zealand, *Mining Meaning from Wikipedia*,[25] describes the unique structural and content reasons why Wikipedia is an absolutely irreplaceable source for notable entities, and semantic Web and natural language research.

DBpedia Ontology	This ontology, an extension of the base DBpedia knowledge base, is meant to be an organizational framework for the information in Wikipedia infoboxes. There are more than 700 types in this ontology.
Dublin Core	Dublin Core, and its metadata extensions, is a generalized vocabulary for describing conceptual works, developed by the library community. It is a widely used core vocabulary across many domains.
Bibliography Ontology	This generalized bibliographic ontology is used to describe books and periodicals; it is the most widely used bibliographic schema.
Description of a Project (DOAP)	A general vocabulary for describing projects.
Friend of a Friend	FOAF is a project devoted to linking people and information using the Web based on social networks, representational networks, and information networks.
FRBR	This vocabulary for the Functional Requirements for Bibliographic Records is a recommendation of the International Federation of Library Associations and Institutions (IFLA) for how to structure catalog databases to reflect the conceptual structure of information resources.
Geo	Geo is a vocabulary for representing latitude, longitude and altitude information in the WGS84 geodetic reference schema.
Music Ontology	MO is a vocabulary for describing music-related topics (i.e., artists, albums and tracks).
Open Organizations	OO is a vocabulary that provides supplementary terms for organizations wishing to publish open data about themselves.
Organization Ontology	The Organization Ontology is a core ontology for organizational structures, aimed at supporting linked data publishing of organizational information across some domains.
Programmes Ontology	The Programmes Ontology is a simple vocabulary for describing media programs. It covers brands, series (seasons), episodes, broadcast events, broadcast services, etc
SIOC	The SIOC initiative (Semantically Interlinked Online Communities) is a vocabulary for the integration of online community information.
Time Ontology	The OWL-Time ontology is a vocabulary of temporal concepts, for describing the temporal properties of resources in the world or described in Web pages.
TRANSIT	TRANSIT is a vocabulary for describing transit systems, routes, stops, and schedules.
US PTO	The US Patent and Trademark Office provide links to millions of organizations and brands that have sought or received trademark protection from the US government.

Table 11-2: Secondary Knowledge Bases for KBpedia

The base KBpedia also includes entities mappings (organizations only) to Freebase (though abandoned, prior users have transferred much Wikidata) and the US Patent and Trademark Office (USPTO) databases. Since these are not full mappings, we do not include them in the statistics for the base KBpedia.

Candidate KBs for Expansion

For specific domains, multiple and rich sources may exist to expand KBpedia to accommodate that scope. *Chapter 13* develops the topics of finding and screening such sources, using some of the acceptance criteria above. For now, let me note that, in varying degrees, vocabularies, thesauri, taxonomies, and, even, tables of content may be useful starting points for domain concepts and scope expansions. One may find local instance data from internal relational datastores and spreadsheets. Sometimes you may find useful domain data and structure from academic publications, trade organizations, or various sector studies.

As for KBpedia, some new areas that we are contemplating include country-specific economic and demographic data, more online databases, brand and product data, expanded corporate and ownership data, sustainability metrics associated with significant economic pathways, or lexical databases, such as WordNet or VerbNet. As a sponsor of the open source project, we will be responsive to multi-lingual versions and will work to catalyze more mappings, more linkages, and more extensions.

Chapter Notes

1. Some material in this chapter was drawn from the author's prior articles at the *AI3:::Adaptive Information* blog: "OWL Ontologies: When Machine Readable is Not Good Enough" (Mar 2006); "An Intrepid Guide to Ontologies" (May 2007); "Ontologies as the 'Engine' for Data-Driven Applications" (Jun 2009); "Confronting Misconceptions with Adaptive Ontologies" (Aug 2009); "Ontology-driven Applications Using Adaptive Ontologies" (Nov 2009); "When Linked Data Rules Fail" (Nov 2009); "An Executive Intro to Ontologies" (Aug 2010); "The Nature of Connectedness on the Web" (Nov 2010); "Ontology-Driven Apps Using Generic Applications" (Mar 2011); "The Age of the Graph" (Aug 2012); "Big Structure: At The Nexus of Knowledge Bases, the Semantic Web and Artificial Intelligence" (Jul 2014); "What is Big Structure?" (Aug 2014); "The Value of Connecting Things - Part I: A Foundation Based on the Network Effect" (Sep 2014); "The Value of Connecting Things - Part II: The Viking Algorithm" (Sep 2014); "The Value of Connecting Things - Part III: Ten Benefits from Big Structure" (Sep 2014); "'Deep Graphs': A New Framework for Network Analysis" (Apr 2016); "Uses and Control of Inferencing in Knowledge Graphs" (Mar 2017); "KBpedia Relations, Part I: Smarter Knowledge Graphs" (May 2017).

2. Topics in discrete mathematics, which are all applicable to graphing techniques and theory, include theoretical computer science, information theory, logic, set theory, combinatorics, probability, number theory, algebra, geometry, topology, discrete calculus or discrete analysis, operations research, game theory, decision theory, utility theory, social choice theory, and all discrete analogues of continuous mathematics.

3. Sylvester, James Joseph, "Chemistry and Algebra," *Nature*, vol. 17, 1878, p. 284.

4. Bunimovich, L. A., Smith, D. C., and Webb, B. Z., "Specialization Models of Network Growth," *arXiv:1712.01788 [physics]*, Dec. 2017.

5. "Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets," is a quote from Steve Lohr, 2014, "For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights," August 17, 2014, New York Times, see http://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html. Also, as another example of the common 80% estimate for data preparation costs, see http://radar.oreilly.com/2013/09/data-analysis-just-one-component-of-the-data-science-workflow.html.

6. These benefits of the network effect were reportedly a major driver of Theodore Newton Vail's efforts to consolidate the thousands of initial telephone networks in the United States under the banner of the Ameri-

can Telephone & Telegraph (Ma Bell) company.

7. Gilder, G., "Metcalfe's Law and Legacy," *Forbes ASAP*, Sep. 1993, p. 158.

8. For a well-connected network, every node (n) connects to every other node (n-1), which gives us n*(n-1) or (n2 – n). Working this out, two nodes have two connections (2*2 – 2), three nodes have six connections (3*3 – 3), and the expression converges on the square of 'n' for larger values of 'n,' e.g., (100*100 – 100) is 99% of (100*100). This convergence at larger number is the basis for the exponential simplification, 2n. Most of the other 'laws' stated herein are simplifications in a similar manner.

9. Peralta, S. F., *Moore's Law, Metcalfe's Law, and the Dot Com Bubble*, 2011.

10. Reed, D. P., "That Sneaky Exponential—Beyond Metcalfe's Law to the Power of Community Building," *Services, in XVth International Symposium on Services and Local Access*, Edinburgh: 1999.

11. Briscoe, B., Odlyzko, A., and Tilly, B., "Metcalfe's Law is Wrong," *IEEE Spectrum*, Jul. 2006.

12. See, for example, https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Zipf_s_law.html.

13. Stein, Y. J., "The Value of Being Linked In" Available: http://www.dspcsp.com/pubs/linkedin.pdf.

14. Ontologies also provide the means for tagging (providing structure) to unstructured documents, which also brings the multiplier benefits from structure. On the retrieval side, such structure also aids faceting and filtered "slicing and dicing" of underlying datasets, thereby improving retrieval efficacy.

15. Harris, S., and Seaborne, A., *SPARQL 1.1 Query Language*, World Wide Web Consortium, 2013.

16. Kuba, M., "OWL 2 and SWRL Tutorial" Available: https://dior.ics.muni.cz/~makub/owl/.

17. McGuinness, D. L., "Ontologies Come of Age," *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, eds., MIT Press, 2003, pp. 171–194.

18. Bodenreider, O., and Olken, F., eds., "Ontology Summit 2007 Communique," *Ontology Summit 2007*, Gaithersburg, MD: Ontolog Forum, 2007.

19. Guarino, N., "Formal Ontology and Information Systems," *Proceedings of FOIS'98*, Trento, Italy: IOS Press, 1998, pp. 3–15.

20. Uschold, M., "Ontology-Driven Information Systems: Past, Present and Future," *Proceedings of the Fifth International Conference on Formal Ontology in Information Systems (FOIS 2008)*, C. Eschenbach and M. Grüninger, eds., Amsterdam, Netherlands: IOS Press, 2008, pp. 3–20.

21. Costa, P. C., "Bayesian Semantics for the Semantic Web," Ph.D., George Mason University, 2005.

22. Laskey, K. B., "MEBN: A Language for First-Order Bayesian Knowledge Bases," *Artificial Intelligence*, vol. 172, Feb. 2008, pp. 140–178.

23. Bobillo, F., and Straccia, U., "Fuzzy Ontology Representation Using OWL 2," *International Journal of Approximate Reasoning*, vol. 52, Oct. 2011, pp. 1073–1094.

24. Suchanek, F. M., and Weikum, G., "Knowledge Bases in the Age of Big Data Analytics," *Proceedings of the VLDB Endowment*, 2014, pp. 1713–1714.

25. Medelyan, O., Milne, D., Legg, C., and Witten, I. H., "Mining Meaning from Wikipedia," *International Journal of Human-Computer Studies*, vol. 67, 2009, pp. 716–754.

26. An exception is the biomedical community through its Open Biological and Biomedical Ontologies (OBO) initiative.