Knowledge Graphs: Introduction, History and Perspectives
Vinay K. Chaudhri, Chaitanya Baru, Naren Chittar, Xin Luna Dong, Michael Genesereth, RV Guha, Jim Hendler, Aditya Kalyanpur, Doug Lenat, Juan Sequeda, Denny Vrandečić, Kuansan Wang

Abstract
Knowledge Graphs (KGs) have emerged as a compelling abstraction for organizing the world’s structured knowledge as well as to integrate information extracted from multiple, heterogeneous data sources. Knowledge graphs have started to play a central role in representing the information extracted by AI systems using natural language processing and computer vision. Domain knowledge expressed in KGs is being input into machine learning models to produce better predictions. The goals of this article are to (a) define KGs and discuss three of their important applications that have gained recent prominence, (b) situate the KGs in the context of the prior work in AI, and (c) present a few contrasting perspectives that help in better understanding KGs in relation to related technologies.

Knowledge Graph Definition
A knowledge graph (KG) is a directed labeled graph in which domain specific meanings are associated with nodes and edges. A node could represent any real-world entity, for example, people, company, computer, etc. An edge label captures the relationship of interest between the two nodes, for example, a friendship relationship between two people, a customer relationship between a company and person, or a network connection between two computers, etc. There are multiple approaches to associate meanings with the nodes and edges. At the simplest level, the meanings could be stated as documentation strings expressed in a human understandable language such as English. At a computational level, the meanings can be expressed in a formal specification language such as the First Order Logic. An active area of current research is to automatically compute the meanings that are captured in a vector consisting of a sequence of numbers. We will contrast these approaches in a later section on symbolic vs vector representations.

Information can be added to a KG via a combination of human-driven, semi-automated, and/or fully automated methods. Regardless of the method, it is expected that the recorded information can be easily understood and verified by humans. We will contrast different approaches to creating a KG in a later section on human curation vs machine curation.

Search and query operations on KGs can be reduced to graph navigation operations. For example, in a friendship KG, to obtain the friends of the friends of a person A, one can navigate the graph from A to all nodes B connected to it by a relation labeled as friend, and then recursively to all nodes C connected by the friend relation to each B. Use of graph algorithms leads to a well-understood efficient computational behavior, and yet it is not sufficient to capture all the inferences of interest. We will discuss this in more detail in a later section on big semantics vs little semantics.

Practical systems adapt the directed labeled graph representation to suit specific application requirements. For example, a KG model prominently used over the World-wide Web, called the Resource Description Framework (RDF) (Brickley et. al, 1999), uses International Resource Identifiers to uniquely identify things. Property graph models (Robinson et. al. 2015) associate properties and values with each node and each edge. Edge properties are useful to represent facts that are in dispute (e.g., country in which Kashmir resides), are highly time-dependent (e.g., the president of USA), or have genuine diversities (e.g., user behaviors). With the recent emphasis on responsible AI, annotating the edges with information on how they were obtained plays a key role to explain the inferences based on the KG. For example, an edge property of confidence may be used to represent the probability with which that relationship is known to be

1 Pending Confirmation.
true. SPARQL (Pérez et al. 2006) and Graph Query Language, query languages supported respectively by the RDF and the property graph models, enable computer programs to query the information in the corresponding KGs.

Recent Applications of Knowledge Graphs

Three important applications that have led to the recent surge in the popularity of KGs are: organizing information over the internet, integrating data in enterprises, and representing information learned by AI algorithms. For the internet KGs, information is available in the open, whereas for enterprise KGs, the information is typically closed, or access controlled. We will highlight new and different issues that may arise in each case.

Organizing Knowledge over the Internet

Wikidata is a collaboratively edited KG that provides data for Wikipedia and for other uses on the web (Vrandečić and Krötzsch, 2014). We will consider an example to illustrate how Wikidata is able to enhance and improve Wikipedia.

The Wikipedia page for Winterthur listed its twin towns: two are in Switzerland, one in Czech Republic, and one in Austria. The city of Ontario in California that has a Wikipedia page titled, Ontario, California, listed Winterthur as its sister city. Sister city and twin city relationships are identical as well as reciprocal. Thus, if a city A is a sister (twin) city of another city B, then B must be a sister (twin) city of A. As “Sister cities” and “Twin towns” are section headings in Wikipedia, with no definition or relationship specified between the two, it is difficult to detect this discrepancy automatically. In contrast, in the Wikidata representation of Winterthur, there is a relationship called twinned administrative body that lists the city of Ontario. As this relationship is defined to be a symmetric relationship in the KG, the Wikidata page for the city of Ontario automatically gets linked to Winterthur. Wikidata solves the problem of identifying inverse relationships through the relation definitions created by curators, and by using constraint checking and inference possible through a KG. To the degree the Wikidata KG is fully integrated into Wikipedia, the discrepancies of missing links considered in the example considered here would naturally disappear. We can visualize the two-way relationship between Winterthur and Ontario in Figure 1. The KG in Figure 1 also shows other objects to which Winterthur and Ontario are connected.

![Figure 1: A fragment of the Wikidata knowledge graph](image)

Wikidata includes data from several independent providers such as the Library of Congress. By using the Wikidata identifier for Winterthur, the information released by the Library of Congress can be easily linked with other information about Winterthur present in Wikidata. Wikidata makes it easy to establish such links.
by publishing the definitions of relationships used via the Schema.Org ontology (Guha, Brickley and McBeth 2016).

A well-documented list of relations in Schema.Org, also known as the relation vocabulary, and now in use by over 35 million web sites, gives us, at least, two advantages. The first advantage is that it is easier to write queries that span across multiple datasets because queries can be framed using relations that are common to those sources (Peng et. al. 2018). Without such common relationships across multiple sources, we would need to determine appropriate translations. An example request that goes across multiple sources is: Display on a map the birth cities of people who died in Winterthur. The second advantage of Schema.Org is that the search engines can use such requests to retrieve information from the KG and display the query results. Use of structured information returned in the search results is now a standard feature for the leading search engines (Noy et. al. 2019).

As of 2021, Wikidata had over 90 million objects, with over one billion relationships among those objects. Wikidata makes connections across over 4872 different catalogs in 414 different languages published by independent data providers. As per a recent estimate, 31% of all websites, and over 12 million data providers are currently using the vocabulary of Schema.Org to publish annotations to their web pages.

What is particularly new and exciting about the Wikidata KG? First, it is a public graph of unprecedented scale, and is one of the largest KGs available today. Second, even though Wikidata is manually curated, the cost of curation is shared by a community of contributors. Third, some of the data in Wikidata may come from automatically extracted information (Wu, Hoffman and Weld, 2008), but it must be easily understood and verified as per the Wikidata editorial policies. Fourth, there is an explicit effort to provide semantic definitions of different relation names through the vocabulary in Schema.Org. Finally, Wikidata is not the only initiative of its kind suggesting the possibility of a massive scaling of this concept. As a concrete example, Data Commons™ performs the necessarily cleaning and joining of data from publicly available government and authoritative sources to distribute a publicly available KG. With this service, the users need not repeat the effort required for data cleaning or joining. Data Commons covers data sources in domains such as demographics (US Census, Eurostat), economics (World Bank, Bureau of Labor Statistics, Bureau of Economic Analysis), health (World Health Organization, Center for Disease Control), Climate (Intergovernmental Panel on Climate Change, National Oceanic and Atmospheric Administration), and many others.

**Integrating data in Enterprises**

Modern enterprises operate in an environment in which the data reside in multiple different databases and other un-structured sources. Many business operations have moved online allowing the capture of the user behavior. In addition, there has been a proliferation of data vendors who are able to provide highly valuable data for businesses. To support efficient business operations, to provide better customer services, and to leverage hidden patterns in the data, efficient and scalable techniques for data integration are central.

Let us consider an example scenario (Ding et. al. 2021). Financial news reports that “Acma Retail Inc” has filed for bankruptcy due to the pandemic, because of which many of its suppliers will face financial stress. For example, if a company A who is a supplier for Acma is undergoing financial stress, a similar stress will be experienced by companies who are suppliers of A. Such supply chain relationships are curated as part of a commercially available dataset called Factset™. In a 360-degree view, the data from Factset and the financial news are integrated with internal customer databases. The resulting KG accurately tracks the Acma supply chain, identifies stressed suppliers with different revenue exposure, and identifies companies whose risk may be worth monitoring.

To create the 360-degree view of a customer, the data integration process begins with knowledge engineers working with business analysts to sketch out a schema of the key entities, events, and the relationships they are interested in tracking (See Figure 2). An essential part of the process is for the users to agree on the meaning of the terms. For example, whether an organization becomes a customer at the time of placing an order, or at the time when the product is delivered? The visual nature of the KG schemas makes it easier for the business users and subject matter experts to specify their requirements. The KG
schema is then mapped to the underlying sources so that the data can be loaded into a KG engine. The meaning of the data stored in enterprises databases is hidden in logic embedded in queries, data models, application code, written documentation or simply in subject matter experts’ minds because of which the mapping process often requires both human and machine effort (Sequeda and Lassila 2021). The storage format of triples allows the system to translate only those relationships that are of immediate relevance to the schema defined by the business domain experts. The rest of the data may still be loaded as triples but without incurring the upfront cost of relating to the defined schema.

What is particularly new and exciting about the use of KGs for data integration? First, the integrated information comes from both textual (e.g., news) as well as structured data sources (e.g., relational databases). As many information extraction systems output information in triples, a generic schema of triples substantially reduces the cost of starting such data integration projects. Second, it is much easier to adapt a triple-based schema in response to changes than the comparable effort required to adapt a traditional relational database. Third, and finally, modern KG engines are highly optimized for answering questions that require traversing the graph relationships in the data. For the example schema of Figure 2, a graph engine has built-in operations to identify the central suppliers in a supply chain network, closely related groups of customers or suppliers, and spheres of influence of different suppliers. All these computations leverage domain-independent graph algorithms such as centrality detection and community detection. Because of the ease of creating and visualizing the schema, and the built-in analytics operations, KGs are becoming a popular solution for turning data into intelligence in the enterprises.

Representing Information for AI Algorithms

We can view KGs at the confluence of two important trends in AI: progress in deep learning and success of hybrid methods. Because of the recent progress in deep learning, natural language processing (NLP) and computer vision (CV) algorithms are starting to move beyond the basic recognition tasks to extracting relationships among objects necessitating a representation in which the extracted relations could be stored for further processing and reasoning. Indeed, the success of hybrid methods in IBM’s Watson (Ferrucci, et. al. 2010) has prompted many to pursue a combination of symbolic and statistical approaches for common sense reasoning. We will next illustrate the use of KGs for NLP, CV, and commonsense reasoning.

From the sentence shown in Figure 3, we can extract the entities Albert Einstein, Germany, Theoretical Physicist, and Theory of Relativity; and the relations born in, occupation and developed. Once this snippet of knowledge is incorporated into a larger KG, we can use logical inference to get additional links (shown by dotted edges) such as a Theoretical Physicist is a kind of Physicist who practices Physics, and that Theory of Relativity is a branch of Physics.
In computer vision, an image is represented as a set of objects with a set of properties each of which corresponds to a bounding box identified by an object detector, and each of the objects is connected by a set of relationships that are detected by an edge detector. For example, from the image shown in Figure 4, a CV algorithm produces the KG shown to the right with objects such as a man, a horse, and a bucket, and relationships such as holding, feeding, etc. In modern CV research, such a KG is referred to as a scene graph (Chen et al. 2019). The scene graphs have become a central tool to achieve compositional behavior in CV algorithms, that is, once a CV algorithm has been trained to recognize certain objects, then by leveraging scene graphs, it can be trained to recognize any combination of those objects with much fewer examples. The scene graphs are also a foundation of tasks such as visual question answering (Zhu et al. 2016).

We next take the example of a specific kind of commonsense reasoning known as cause effect reasoning. Given an event such as X repels Y’s attack, humans can make many commonsense inferences about why did the repel happen? How does X feel about the attack? What might be likely effect of such a repel? A general strategy to program such reasoning is to first curate a KG manually, such as the one shown in Figure 5, and then use that data in conjunction with a machine learning algorithm to predict the cause and effects for events that do not exist in the KG. For example, given a new event such as X leaves without Y, the system will make inference such as X wanted to be alone, X wanted to go home, Y might miss his friend,
Two examples of such systems are ATOMIC that contains over 300,000 event nodes and over 800,000 cause effect triples (Sap et. al. 2019), and Glucose that contains over 670,000 cause effect triples (Mostafazadeh, et. al. 2021).

Let us now consider what is new and exciting about these recent uses of KGs for NLP, CV and commonsense reasoning. First, and foremost, automated creation is a central component to the approach. For the commonsense reasoning KGs, even though there is a significant upfront manual effort to create the training set, but once trained, the learning algorithm can deal with new cases. Second, there is a clear recognition that KG representations are a central ingredient to achieving the compositional behavior in AI systems. Even though this recognition is expressed most vocally in the context of scene graph, but a similar assumption is implicit for capturing the output of NLP, and in the rationale for creating cause effect KGs.

**Prior Research Related to Knowledge Graphs**

AI agents have the need to maintain representations of the real world—or, for that matter, any world that they may be simulating—and use this representation for reasoning. Choosing the ideal, or even a good, representation that allows AI agents to store information and derive new conclusions from it is a problem that is central to AI. The current work and interest in KGs can be situated in this broader context of AI.

Earliest research in AI used frame representations, known as semantic networks, that consisted of nodes and labeled edges (Woods 1975). This directed labeled graph representation has been adapted depending on the needs of an application and given new names. For example, a directed labeled graph where the nodes are, say, people, and the edges capture the parent relationship is sometimes referred to as a relational structure. A directed labeled graph where the nodes are classes of objects (e.g., Book, Textbook, etc.), and the edges capture the subclass relationship, is known as a taxonomy. In some data models, given a triple (A, B, C), we refer to A, B, C as the subject, the predicate, and the object of the triple respectively. For example, given the triple (“Biden”, “president”, “USA”), “Biden” is the subject, “president” is the predicate, and “USA” is the object of the triple. A labeled graph that combines data and taxonomy is often referred to as an ontology.

While some researchers used First Order Logic to computationally understand semantic networks (Hayes 1981), others advocated that at least First Order Logic was required to represent the knowledge needed for AI agents (McCarthy 1979). Because of the computational difficulty of reasoning with First Order Logic, different subsets of First Order Logic such as the description logics (Brachman and Lévesque 1984) and the logic programs (Kowalski 1979) were investigated. There was an analogous development in databases where the initial systems were based on a network data model (Taylor and Frank, 1976). But a desire to achieve independence between the data model and the query processing led to the development of relational data model (Codd, 1983) that shares its mathematical core with logic programming.

The foundational research was accompanied by implemented KR systems. For example, the representation system CycL (Lenat and Guha) combined ideas from the First Order Logic and frame representation systems in the context of the practical requirements of coding knowledge on a spectrum of topics (Lenat 1995). These early systems were used to capture the knowledge of an intelligent agent, including the rules of causality, implications of relationships between entities, commonsense rules, expert rules, etc. This trajectory of development in AI can be loosely characterized as starting from the need for explicit representations (McCarthy 1979, Newell 1982) to expert systems (Feigenbaum, 1984) to large common sense knowledge bases (Lenat 1995). These systems had complex axioms with sophisticated inference mechanisms, but the overall scale, measured in terms of the number of axioms, has been relatively small. The goal was to use the rules to model human reasoning.

Mid-nineties saw an explosion of information on the web, and better methods to access and search this information were needed. There was tremendous success in using information retrieval methods such as the Page rank algorithm (Page et. el. 1999), and yet it was felt that more was possible if there was a way for us to convey the semantics to our search algorithms (Berners-Lee et. al. 2001). That vision has come to fruition with the improvement in the search results with resources such as Wikidata and Data Commons.
Both Wikidata and Data Commons use representations heavily influenced by an earlier language called the Meta Content Format (Guha 1996) and CycL. In contrast to the early AI systems, the KGs on the world-wide web emphasize capturing many ground facts that are used in applications such as search and analytics with much less emphasis on complex inference. A broader account of the historical developments of KGs outside AI is available elsewhere (Gutiérrez and Sequeda 2021).

**Contrasting Perspectives**

Three contrasting perspectives have emerged with the increasing adoption and use of KGs in different scenarios and use cases: symbolic representation vs vector representation, human curation vs machine curation, and “little semantics” vs “big semantics.” There are spirited debates in the community about the effectiveness and efficacy—sometimes even the validity! —of each approach, with the adherents of one perspective sometimes claiming “superiority” of their approach over the other. Our goal in presenting the differing perspectives here is to enable better understanding of each and articulate the problems where a solution of a certain kind is appropriate.

**Symbolic Representation vs Vector Representation**

Machine learning algorithms used for NLP and CV rely on a vector representation of text and images. The recent success of deep learning on multiple tasks has prompted many to reject the need for any symbolic representation. We will examine these alternative views more closely.

A commonly used vector representation in NLP is known as word embeddings. For example, given a corpus of text, we can count how often a word appears next to every other word giving us a vector of numbers. Sophisticated algorithms exist for calculating compact word embeddings (Mikolov et. al. 2013). Word embeddings are then leveraged in tasks such as word similarity calculation, entity extraction, and relation extraction. Analogously, the CV algorithms operate on vector representation of images.

Vector representations have excelled at many tasks including web search. Such methods have no difficulty answering a question such as: Who was the prime minister of the UK in October of 1956? These methods fail miserably (Lenat 2019a, Lenat 2019b) if we modify the question to: Who is the prime minister of the U.K. when Theresa May was born? Humans have little difficulty understanding and answering these questions. We can address the limitations of vector representations by encoding the information extracted from text and images into a KG as we saw in Figures 3 and 4. Complementing the vector and symbolic representations enables the programs to achieve compositional behavior and facilitates further inference and reasoning. Graph embedding is a generalization of word embedding but for graph structured input (Hamilton 2020). Use of graph embeddings with a neural network are being used for handling unseen actions in the cause effect KGs we considered earlier.

Neuro-symbolic reasoning is a fast-emerging area of research that leverages the benefits of automatic calculation of embeddings, and yet, recognizes the need for a discrete KG to produce a human understandable representation. We illustrate neuro symbolic reasoning on a story understanding task (Dunietz et. al. 2020). Consider the following story: *Fernando went to a plant shop. He liked the minty smell of the leaves. He bought a plant and placed it next to a window*. Given this story we want to answer the question: *Why did Fernando buy the plant?* A possible human understandable chain of reasoning to answer this question involves the following steps: (a) If A has part B, and B has property P then A has property P; (b) If A likes property P of B, then A likes B; (c) If A likes B, A may buy B. In this chain of reasoning, steps (a) and (b) are examples of the rules that may exist in a traditional symbolic knowledge base, whereas (c) is a probabilistic rule of the sort that we may find in a cause effect KG that we considered in the earlier section. Such rules may already exist as part of the curated portion of the KG or could be inferred ahead of time using a graph neural network or could be inferred dynamically in response to a query. A neuro symbolic reasoner can manage and execute this reasoning process (Kalyanpur et. al. 2020).
Human Curation vs Machine Curation

Industrial KGs, for example, the Google KG, Amazon Product Graph (APG), and Microsoft Academic Graph (MAG) are of unprecedented scale (Noy et al. 2019). There has often been debate on the degree to which we could create such KGs exclusively through automated methods (also referred to as machine curation) vs creation through human effort. We will illustrate this tradeoff through two examples based on the Microsoft Academic Graph and Amazon Product Graph that leveraged significant automation, and two examples based on the Wikidata KG and the Cyc knowledge base (Lenat 1995) which were primarily created through human curation.

The MAG team faced the problem of uniquely identifying authors and their publications. A human curation strategy advocates setting up standards such as Document Object Identifier (DOI) for uniquely identifying publications, and Open Researcher and Contributor ID (ORCID) for uniquely identifying authors. These standards rely on the authors and publishing organizations to contribute manual efforts to annotate documents with DOIs and ORCIDs. Human curation of even such simple principles has been problematic for several reasons. First, such identifiers have had low human readability. Second, frequent typographical errors have created an adoption barrier. Third, not having DOIs for the publications has not hampered their accessibility as there are multiple ways to find publications on the web. Finally, there is some abuse of the uniform identifiers. For example, some individuals acquire multiple identifiers to partition their publications into separate profiles defeating the design goal of ORCID being a unique identifier. The MAG team consequently leveraged machine curation as it is possible to identify a publication by its contents and an author can be disambiguated based on the fields of research, affiliations, coauthors, and other factors that are more natural to humans.

The APG contains millions of categories of products, thousands of attributes of those products, which are expressed in hundreds of languages. Even though one may assume that the vendors wishing to sell their products would volunteer information that could be directly input into the APG, but in practice, that was not found to be the case. Creating the APG entirely through human curation would have required hundreds of person years of effort. Hence different machine curation techniques needed to be leveraged at different levels of scaling. To get the project off the ground, highly accurate automated knowledge extraction models were created through manual effort (Zheng et al. 2018). The next level of scaling required leveraging AutoML and automatic cleaning techniques (Wang et al. 2020) so that each knowledge extraction model need not be manually tuned and optimized. Scaling even more required reducing the total number of models required for the variety of knowledge to be extracted which was achieved through transfer learning techniques (Karamanolakis et al. 2020). The final level of scaling required using multimodal information (for example, extraction from text as well as images) (Lin et al. 2021, Yan et al. 2021). Human-created and highly precise models were the foundation of this process. Different levels of scaling, however, required leveraging techniques such as named entity recognition, closed information extraction, knowledge cleaning, and knowledge-based question answering.

The Wikidata KG that we considered earlier in the paper was launched to solve the problem that the data in Wikipedia is buried in 30 million articles in 287 different languages from which automatic extraction is inherently difficult. The same information often appears in articles in many languages and in many articles within a single language. Population numbers for Rome, for example, can be found in English and Italian articles about Rome but also in the English article “Cities in Italy.” The numbers are all different. Wikidata is founded on the principle of plurality that it would be naive to expect global agreement on the “true” data, since many facts are disputed or simply uncertain. Unlike, MAG and APG, Wikidata allows conflicting data to coexist and provides mechanisms to organize this plurality. Checking, verifying, and allowing such plurality requires human curation and cannot be left to machines. Wikidata has been created through human curation with over 20,000 active editors and a community of over 400,000 editors. Because of the strength of its editor community, Wikidata has leveraged standard published identifiers including the
International Standard Name Identifier (ISNI), China Academic Library and Information System (CALIS), International Air Transport Association (IATA), MusicBrainz for albums and performers, and North Atlantic Basin’s Hurricane Database (HURDAT). Wikidata itself publishes standard identifiers for items that appear in it that are now increasingly being used in commercial KGs.

Finally, the Cyc knowledge base, which is the largest and most complex model of human common sense that exists today, was largely created through human curation. This was necessary because the project aims to capture knowledge, which is not explicitly written down in the text, and hence, cannot be automatically extracted through the methods of NLP and CV. Early versions of Cyc (circa 1984) used representations like present day KGs, but since 1989 Cyc has used higher order logic as its knowledge representation language CycL, which for example, includes nested modals, as it needs to represent – let alone correctly reason about answers to queries like: When Juliet drank her potion, what did she expect that Romeo would believe once he heard that she was dead, and why (Lenat 2019a)? Automatically extracting knowledge into such highly expressive languages is out of the reach of automated techniques even if the knowledge to be entered had been explicitly written down.

In anticipation that sooner or later (and mostly sooner) everyone using KGs will need to move to more expressive representations, Cycorp, the company building Cyc, is working to build Cyc-powered increasingly automatic “power tools” to lower the bar for others to create and modify Cyc applications. They are also setting up a not-for-profit Knowledge Axiomatization Institute “KNAX” which will cast a broad net nationwide and worldwide to identify individuals at any level of education or professional training who have latent talent for “ontological engineering”, and train them in the required skills to contribute to knowledge bases that employ representations richer than KGs.

Little Semantics vs Big Semantics

The big semantics perspective advocates capturing complete meaning of concepts. In contrast, the little semantics perspective emphasizes capturing many facts in a KG. We will illustrate these perspectives by taking examples.

A directed labeled graph representation has its inherent limitations in capturing the big semantics of an application. A simple example of such a limitation is to capture a statement such as: Los Angeles is between San Diego and San Jose along US 101. We can capture this statement in a directed labeled graph using a technique known as reification, but it requires multiple triples (see Figure 6a). This statement can be captured in a single statement if we allow four-place predicates which are not supported in the pure
model of directed graphs (although many implementations of graph and semantic web databases include these). For this example, the KG representation is akin to using the assembly language as opposed to a higher-level programming language. Furthermore, it makes downstream tasks such as natural language generation more difficult which now must assemble information spread across multiple triples. As a more involved example, consider the statements Every Swede has a King, and Every Swede has a mother, which are syntactically similar in English, and many KGs would represent them identically, but these statements have very different computational meaning (see Figure 6b). Because of the need to capture the full semantics in machine reasoning, CycL was designed to be even more expressive than the First Order Logic.

Despite the limitations of the directed labeled graph representation, it has been found useful for solving many practical problems that require little semantics. Wikidata, Data Commons, MAG and APG all use a directed labeled graph representation at their core, and their existence and commercial usefulness is a strong evidence that a little semantics goes a long way (Hendler, 2007). Furthermore, even for the simple directed labeled graph representation, there are numerous unsolved problems. For example, how might we create open KGs? (This is precisely the question that is being addressed by multiple Open Knowledge Network projects in this special issue.) What common naming conventions will allow users to interact with multiple existing KGs and create their own combined products which in turn can be used by others and combined still further, ad infinitum? How do we support co-designation of objects in different KGs, i.e., which objects reference the same real object? Even though these problems may seem small, good solutions to them will be critical in advancing our ability to create open KGs.

Summary and Conclusion

<table>
<thead>
<tr>
<th></th>
<th>Semantic Networks</th>
<th>Knowledge Graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scale</strong></td>
<td>Thousands of objects</td>
<td>Billions of objects</td>
</tr>
<tr>
<td></td>
<td>Complex logical inference</td>
<td>Scalable graph algorithms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Neuro Symbolic reasoning</td>
</tr>
<tr>
<td><strong>Development Methods</strong></td>
<td>Top-down design</td>
<td>Bottom-up design</td>
</tr>
<tr>
<td></td>
<td>Complex rules and ontologies</td>
<td>Triangles and embeddings</td>
</tr>
<tr>
<td><strong>Modes of Construction</strong></td>
<td>Knowledge engineering</td>
<td>Knowledge engineering,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>crowdsourcing, machine learning</td>
</tr>
</tbody>
</table>

Table 1: How are Knowledge Graphs different from Semantic Networks?

A directed labeled graph is a fundamental construct in discrete mathematics and has applications in all areas of computer science. Most notable uses of directed labeled graphs in AI and databases have taken the form of data graphs, taxonomies, and ontologies. Traditionally, such applications have been small and have been created by a top down design and through manual knowledge engineering.

Distinguishing characteristics of the modern KGs from the classical KGs are scale, bottom-up development, and multiple modes of construction. These differences are also summarized in Table 1. The early semantic networks in AI never reached the size and scale of the KGs that we see today. The emphasis in the early AI systems was also on complex logical inference in contrast to the analytics operations supported by the KGs of today. Difficulty in coming up with a top-down schema design for data integration and the data driven nature of machine learning have forced a bottom-up methodology for creating the KGs. Finally, for creating modern KGs, we are supplementing manual knowledge engineering techniques crowdsourcing and significant automation that is now possible through progress in machine learning.

We would like to conclude by noting that it is not necessary for us to settle the debates on symbolic representation vs vector representation, manual curation vs machine curation and little semantics vs big semantics. It is, however, necessary for us to try many different approaches in parallel and to explore means of combining the various approaches. For now, setting a use-inspired context is a promising approach for
choosing which approaches to explore in both commercial and academic settings. This is well-illustrated by the framework adopted by the projects being reported in the current special issue.

References


Morgan Kaufmann.


DB. Lenat. 2019b. Not As Good As Gold: Today’s AI are Dangerously Lacking in AU (Artificial Understanding), Forbes, February 18, 2019. Available online as


Wang, Y., Xu, Y.E., Li, X., Dong, X.L. and Gao, J., 2020, August. Automatic Validation of Textual Attribute Values in E-commerce Catalog by Learning with Limited Labeled Data. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2533-2541).


Zheng, G., Mukherjee, S., Dong, X.L. and Li, F., 2018, July. Opentag: Open attribute value extraction from product profiles. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1049-1058).

End Notes

i https://www.gqlstandards.org/
ii https://en.wikipedia.org/wiki/Winterthur
iii https://en.wikipedia.org/wiki/Ontario,_California
iv https://www.wikidata.org/wiki/Q9125
v https://www.wikidata.org/wiki/Q488134
vi https://id.loc.gov/authorities/names/n50013808.html?
vii http://datacommons.org
viii http://factset.com
ix https://nlp.stanford.edu/projects/glove/