Received: 23 September 2021 | Accepted: 25 January 2022

DOI: 10.1002/aaai.12033

ARTICLE

Knowledge graphs: Introduction, history, and perspectives

Vinay K. Chaudhri[1] | **Chaitanya Baru**[2] | **Naren Chittar**[3] | **Xin Luna Dong**[4] | **Michael Genesereth**[1] | **James Hendler**[5] | **Aditya Kalyanpur**[6] | **Douglas B. Lenat**[7] | **Juan Sequeda**[8] | **Denny Vrandečić**[9] | **Kuansan Wang**[10]

[1]Stanford University, Stanford, California, USA

[2]San Diego Supercomputer Center, UC San Diego

[3]JPMorgan Chase & Co.

[4]Meta AR/VR Assistant

[5]Renssalear Polytechnic Institute

[6]Elemental Cognition

[7]Cycorp

[8]data.world

[9]Wikimedia Foundation

[10]Microsoft

Correspondence
Vinay K. Chaudhri, JPMorgan Chase & Co. 310 University Ave, Palo Alto, CA 94301, USA.
Email: Vinay.Chaudhri@jpmchase.com

Vinay K. Chaudhri is currently an executive director with JPMorgan Chase & Co.

Funding information
National Science Foundation

Abstract

Knowledge graphs (KGs) have emerged as a compelling abstraction for organizing the world's structured knowledge and for integrating information extracted from multiple data sources. They are also beginning to play a central role in representing information extracted by AI systems, and for improving the predictions of AI systems by giving them knowledge expressed in KGs as input. The goals of this article are to (a) introduce KGs and discuss important areas of application that have gained recent prominence; (b) situate 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.

INTRODUCTION

The term *knowledge graph* (KG) has gained several different meanings across a range of usage scenarios. This paper focuses on the use of KGs in the context of two important current trends: the desire and need to harness the large and diverse data that are now available and the advent of new machine learning capabilities for extracting meaning from unstructured text and images. It provides the authors' perspective on this area and tracks recent efforts in the NSF Convergence Accelerator Track A on Open Knowledge Network (OKN), where the first author was a participant in one of the projects (Baru el al. 2022). All coauthors were speakers in a graduate seminar on KGs at Stanford University, coorganized by the first author, which featured presentations by over 50 speakers[i]. This article strives to provide a synthesis of those diverse perspectives—rather than being an exhaustive survey of the topic area.

KNOWLEDGE GRAPH DEFINITION

A 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*, *companies*, and *computers*. 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.

There are multiple approaches for associating 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 first-order logic. An active area of current research is to automatically compute the meanings captured in a vector consisting of a sequence of numbers. We will contrast these approaches for capturing meaning in a later section on *symbolic versus vector* representations.

Information can be added to a KG via a combination of human-driven, semiautomated, 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 versus machine curation*.

Search and query operations on KGs can be reduced to graph navigation. For example, in a *friendship* KG, to obtain the friends of the friends of a person A, one can first navigate the graph from A to all nodes B connected to it by a relation labeled as *friend*. One can then recursively navigate to all nodes C connected by the *friend* relation to each B. Directed labeled graph representation and graph algorithms are effective for several classes of problems. They are, however, insufficient to capture all inferences of interest. We will discuss this in more detail in a later section on *big semantics versus 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)* (Cygniak, Wood, and Lanthaler 2014), uses International Resource Identifiers (IRIs) to uniquely identify "things" (entities). *Property graph* models (Robinson, Webber, and Eifrem 2015) associate properties and values with each node and each edge. Edge properties can be used for a variety of purposes: to represent facts that are in dispute (for example, a country in which Kashmir resides); highly time-dependent information (for example, the president of USA); or genuine diversities (for example, user behaviors). With the recent emphasis on responsible AI, annotating the edges with information on how they were obtained plays a key role in explaining inferences based on the KG. For example, an edge property of *confidence* could be used to represent the probability with which that relationship is known to be true. Finally, query languages, such as SPARQL (Pérez et al. 2006) for RDF and Graph Query Language[ii] for property graph models, provide the ability to query the information in respectively RDF and property graph KGs.

APPLICATIONS OF KNOWLEDGE GRAPHS

Two key applications that have led to a surge in popularity of KGs are: (1) integration and organization of information about known "entities," either as an openly accessible resource on the web[iii], or as a proprietary resource within an enterprise/organization; and (2) representation of input and output information for AI/ML algorithms. These application use cases are explored further in the following sections.

Organizing open information

Wikidata is a collaboratively edited open KG that provides data for Wikipedia and for other uses on the web (Vrandečić and Krötzsch 2014). As illustrated in the following example, the Wikidata KG can help enhance and improve the quality of information in Wikipedia. Consider the Wikipedia page for the town, Winterthur[iv], which includes a list of all of Winterthur's *twin towns*: two are in Switzerland, one in the Czech Republic, and one in Austria. Wikipedia also has an entry for the city, Ontario, in California[v], which lists Winterthur as its *sister city*. The "sister city" and "twin city" relationships are meant to be identical as well as reciprocal. Thus, if a city A is a sister (twin) of another city B, then B must be a sister (twin) of A. In Wikipedia, "Sister cities" and "Twin towns" are simply section headings without any relationship/linkage specified between the two. Therefore, it is difficult to detect this discrepancy automatically. In contrast, the Wikidata representation of Winterthur[vi] includes a relationship called *twinned administrative body*, which includes the city of Ontario, CA. As this relationship is defined to be a symmetrical relationship in the KG, a SPARQL query engine can infer that the Wikidata page for the city of Ontario, CA[vii] is to be linked to the Wikidata page of Winterthur.

Wikidata solves the problem of identifying inverse relationships through the relation definitions created by curators and by using inference made possible through a KG inference engine. More advanced forms of such inference are illustrated in the Environmental Intelligence OKN (Janowicz et al. 2022) and the flood impact evaluation OKN

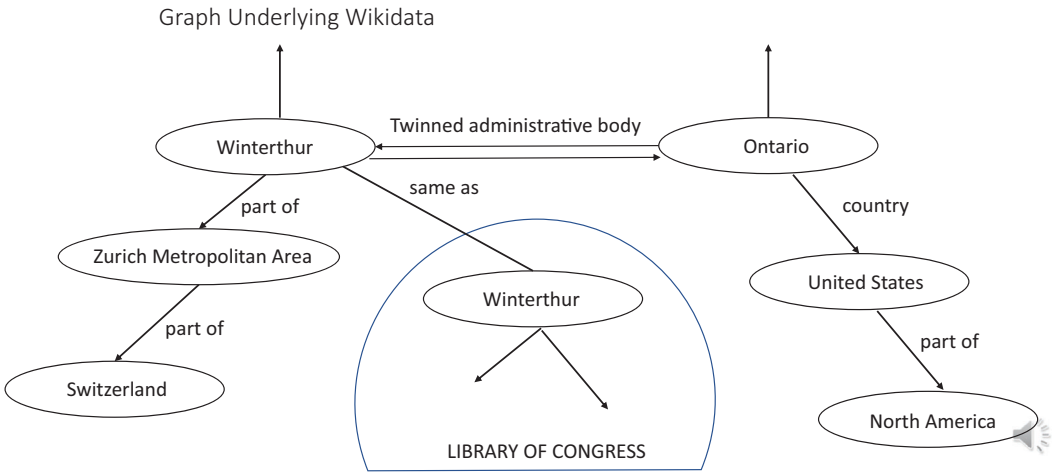

FIGURE 1 A fragment of the Wikidata knowledge graph

(Johnson et al. 2022) reported in this issue. To the degree that the Wikidata KG is fully integrated into Wikipedia, the discrepancy of missing links in the example provided here would not be present. Figure 1 depicts the two-way relationship between Winterthur and Ontario and shows some of the other objects to which Winterthur and Ontario are connected.

Wikidata includes information from several independent providers including, for example, the Library of Congress[viii]. By using unique internal identifiers for distinct entities, for example, Winterthur, from a variety of sources, such as, the Library of Congress and others, the information about an entity can be easily linked together. Wikidata makes it easy to integrate the different data sources by publishing a mapping of the Wikidata relations to the *schema.org* ontology. Such tools were recently leveraged to add information about COVID 19 to Wikidata (Waagmeester et al. 2021). Mappings from relation names in Wikidata to relation names in other sources enable formulation and processing of queries spanning multiple datasets across such sites using relations that are common to that set of sites (Peng et al. 2018). An example of such a request is: *Display on a map the birth cities of people who died in Winterthur*. Without a common relation vocabulary, for example, *birth city*, it would be necessary to create appropriate translations between relations used in one site to the relations used in other sites. Search engines are routinely using the results of such queries to enhance their results (Noy et al. 2019).

As of 2021, Wikidata contained over 90 million distinct 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 (Guha, Brickley, and Macbeth 2016).

There are many new and exciting aspects of the Wikidata KG. First, it is a public graph of unprecedented scale, and one of the largest KGs openly available today. Second, even though it is manually curated, the cost of curation is shared by a community of contributors. Third, while some of the data in Wikidata may be automatically extracted from sources (Wu, Hoffmann, and Weld 2008), all information is required to be easily understandable and verifiable as per the Wikidata editorial policies. Lastly, and importantly, there is a commitment to providing semantic definitions of relation names through the vocabulary in *schema.org*.

A recent example of another openly accessible KG is from the Data Commons[ix] effort whose goal is to make publicly available data readily accessible and usable. Data Commons performs the necessary cleaning and joining of data from a variety of publicly available government and other authoritative data sources and provides access to the resulting KG. It currently incorporates data on 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 sustainability.

Organizing enterprise information

Data integration is essential to the functioning of modern enterprises where corporate data typically reside across many distinct databases and unstructured sources. Furthermore, the broad shift to online operations for almost all enterprises has resulted in the accumulation of very

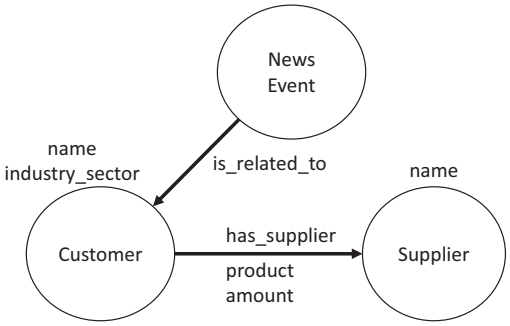

FIGURE 2 An example schema for a 360-degree view

large amounts of valuable user behavior data across distributed locations. In addition, a proliferation of data available from third-party data vendors is providing enterprises highly valuable information which needs to be integrated with internal data for more effective business operations.

Consider the following example: a financial news report has been released stating that "Acma Retail Inc" has filed for bankruptcy due to the pandemic because of which many of its suppliers will face financial stress (Ding et al. 2021). If company C, that is a supplier to Acma, is undergoing financial stress, one might expect that a similar stress is also experienced, in turn, by suppliers to C. Such supply chain relationships are currently being curated as part of a commercially available dataset called Factset[x].

A "360-degree view" of a customer of a company includes the data about that customer from within the company and the data about the customer from sources outside the company. A company could create a "360-degree view" of its customers by combining third-party data, for example, Factset and information from the open financial news with the company's own internal databases. This often requires solving the *entity disambiguation* problem to uniquely identify entities under question—which is also a problem being addressed in the OKN-related projects described in (Cafarella et al. 2022) and (Pah et al. 2022) in this special issue. The resulting KG could be used to track the Acma supply chain and help identify stressed suppliers whose risk may be worth monitoring.

The data integration process for creating the 360-degree view of a customer might begin with knowledge engineers working with business analysts to sketch out a schema of the key entities, events, and the relationships that they are interested in tracking (see Figure 2). An essential part of this process is for the users to agree on the meanings of the terms. For example, when does an "organization" become a "customer"—at the time of placing an order, or at the time when the product is delivered? In practice, the visual nature of the graph-oriented KG schemas facilitates whiteboarding of the schemas by the business users and subject matter experts in specifying their requirements. Next, the KG schema needs to be mapped to the schemas of the underlying sources so that the respective data can be loaded into the KG engine. The meaning of the data stored in enterprise databases is hidden in logic embedded in queries, data models, application code, written documentation, or simply in the minds of subject matter experts requiring both human and machine effort in the mapping process (Sequeda and Lassila 2021).

Let us consider new and exciting aspects of the use of KGs for data integration. First, the integrated information may come from text and other unstructured sources (for example, news, social media, and others) as well as structured data sources (for example, relational databases). As many information extraction systems already output information in triples, using a generic schema of triples substantially reduces the cost of starting such data integration projects. Second, it can be easier to adapt a triple-based schema in response to changes than the comparable effort required to adapt a traditional relational database. This is because a relational system is typically modeled to support the application (McComb 2018), and thus, schema changes often require database reorganization. On the other hand, in a KG system, the schema is modeled to represent the enterprise (McComb 2019), and its representation in triples remains fixed. Lastly, 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 typical graph engine would be able to employ built-in operations for identifying (1) the central suppliers in a supply chain network, (2) closely related groups of customers or suppliers, and (3) spheres of influence of different suppliers. All these computations leverage domain-independent graph algorithms such as centrality detection and community detection.

Due to the relative ease of creating and visualizing the schema and the availability of built-in analytics operations, KGs are becoming a popular solution for turning data into intelligence in the enterprises. For example, the precision medicine OKN reported later in this special volume makes an extensive use of the graph-based visualization and inference for solving problems in biomedicine (Baranzini et al. 2022).

Representing information for AI algorithms

KGs are an essential technology for natural language processing (NLP), computer vision (CV), and commonsense reasoning. As a result of recent advances in deep learning for NLP and CV, algorithms in these domains are moving beyond basic recognition tasks to extracting relationships among objects, thereby requiring a representation scheme

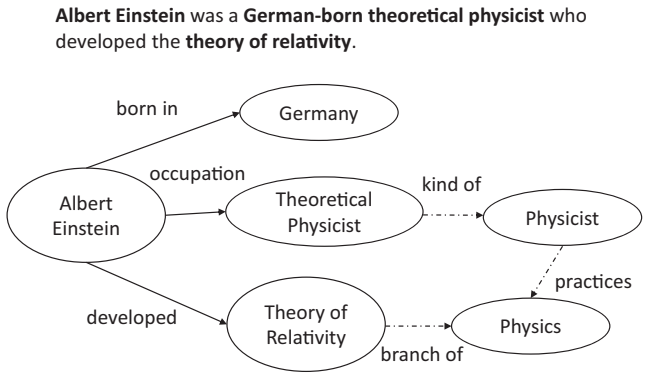

FIGURE 3 A knowledge graph created using entity and relation extraction

in which the extracted relations could be stored for further processing and reasoning. In commonsense reasoning, the success of hybrid methods employed in IBM's Watson (Ferrucci et al. 2010) has prompted many to pursue a combination of symbolic and statistical approaches for common sense reasoning that requires the use of KGs.

Figure 3 depicts an example of the use of KGs to represent knowledge extracted by NLP. It shows a sentence from which one 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 derive 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. The court records OKN project described in this special issue makes an extensive use of similar entity extraction techniques (Pah et al. 2022).

In CV, an image is represented as a set of objects with a set of properties, where each object corresponds to a bounding box, identified by an object detector, and the objects are interconnected by a set of named relationships that are predicted by a model trained for identifying visual relationships. In Figure 4, a CV algorithm produces the KG shown to the right with objects such as *a woman*, *a cow*, and *a mask*, and relationships such as *holding*, *feeding*, and others. In modern CV research, such a KG is referred to as a *scene graph* (Chen et al. 2019), which has become a central tool for achieving 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 fewer examples. Scene graphs also provide the foundation for 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-and-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 the likely effect of such a repel? A general strategy to program such reasoning is to first curate a KG manually and then use it in conjunction with a machine learning algorithm to predict the effects for events that do not exist in the KG. For example, given a new event such as *X leaving without Y*, the system makes inference such as *X wanting to be alone*, *X wanting to go home*, *Y might miss his friend*, etc. 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. 2020).

In these uses of KGs in AI, automated creation of the KG is a central component of the approach. For the commonsense reasoning KGs, even though there is a significant upfront manual effort to create the training set, once trained, the learning algorithm would deal with many new cases at no additional cost. Second, there is a clear recognition that KG representations are a central ingredient to achieving the compositional behavior in AI systems. This is clearly illustrated in the context of a scene graph, but also in capturing the output of NLP and in the rationale for creating cause-effect KGs.

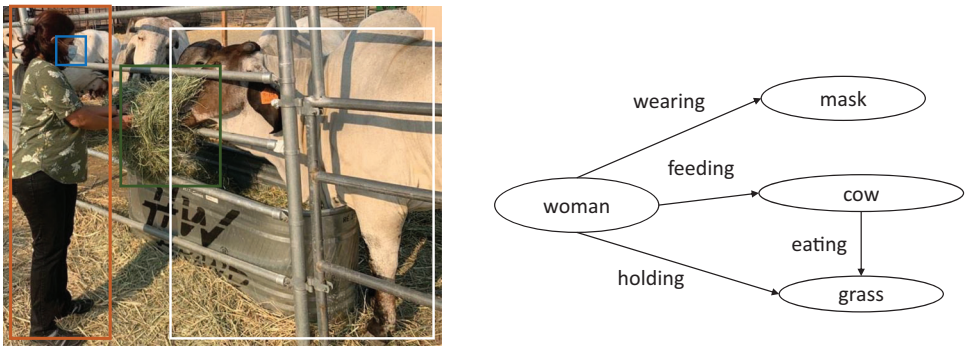

FIGURE 4 A knowledge graph created using computer vision techniques

PRIOR RESEARCH RELATED TO KNOWLEDGE GRAPHS

Graph-based representations of data are employed widely throughout computer science (Borgida and Mylopoulos 2009). AI agents maintain representations of real/simulated worlds and utilize these representations for reasoning in the domain. Indeed, choosing representations that allow agents to store information and derive new conclusions is a problem that is central to AI.

The earliest research in AI used frame representations, known as *semantic networks*, which were directed labeled graphs (Woods 1975). This directed labeled graph representation has been adapted depending on the needs of a given application. 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 (for example, Book, Textbook, and others), 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 directed labeled graph containing data and taxonomy is often referred to as an *ontology*.

While some researchers used first-order logic (FOL) to computationally understand semantic networks (Hayes 1981), others advocated that FOL was required to represent the knowledge needed for AI agents (McCarthy 1989). Because of the computational difficulty of reasoning with FOL, different subsets of FOL, such as description logics (Brachman and Levesque 1984) and logic programs (Kowalski 2014), were investigated. There was an analogous development in databases where the initial data 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 eventually led to the development of relational data model (Codd 1982), which shares its mathematical core with logic programming. A need to handle semistructured data (Buneman 1997) inspired the investigation of "schema-free" systems or triple stores that capture an important class of problems addressed by modern KG systems.

Implemented KR systems accompanied the foundational research. For example, the representation system CycL (Lenat and Guha 1991) combined ideas from FOL and semantic networks 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, and others. This trajectory of development in AI can be loosely characterized as starting from the need for explicit representations (McCarthy 1989; 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.

The mid-1990s saw an explosion of information on the web, and better methods to access and search this information were needed. There was a tremendous success in using information retrieval methods such as the Page Rank algorithm (Page et al. 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, Hendler, and Lassila 2001). That vision is coming to fruition with the improvement in search results with the help of resources such as Wikidata and Data Commons which use representations heavily influenced by an earlier language called the Meta Content Format (Guha 1996). In contrast to the early AI systems, today's KGs 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).

Table 1 describes KG models currently being used by the OKN projects described in this special issue. These include RDF and property graph data models, as well as key-value representation in JSON, and mapping of data into a relational database through suitable translations. Each project addresses semantics either through the development of new ontologies or through leveraging existing ontologies.

CONTRASTING PERSPECTIVES

With the increasing adoption and use of KGs in different scenarios and use cases, three contrasting perspectives have emerged: symbolic representation versus vector representation, human curation versus machine curation, and "little semantics" versus "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 claiming superiority of their approach over the other. Given the breadth of potential applications, it is not necessary for us to settle these debates, but it is important to try many different approaches in parallel and explore

TABLE 1 OKN projects covered in this special issue

OKN project	Representation used	Technical problems addressed
Environmental Intelligence (Janowicz et al. 2022)	RDF	Spatial knowledge, n-degree property path queries, modular ontology development
Flood Impact Evaluation (Johnson et al. 2022)	RDF	Ontology-based inference, multiple related graphs
Precision Medicine (Baranzini et al. 2022)	Property graphs	Graph-inference, network visualization, biomedicine ontologies
Infrastructure for OKNs (Cafarella et al. 2022)	JSON	Author disambiguation, data refinement, data life cycle
Court Records (Pah et al. 2022)	Object relational mapping to a SQL database	Entity disambiguation, ontology of litigation events

OKN, Open Knowledge Network; RDF, Resource Description Framework.

means of combining various approaches to advantage. Our objective in presenting the differing perspectives here is to enable a better understanding of each and articulate the problems where a solution of a certain kind is appropriate.

Symbolic representation versus 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 *word embedding*[xi]. For example, given a corpus of text, one can count how often a word appears next to every other word, resulting in a vector of numbers. Sophisticated algorithms are available for reducing the dimensions of the vectors to calculate a more compact vector, known as a *word embedding* (Mikolov et al. 2013). Word embeddings capture the semantic meaning of the word in a way that can be computationally leveraged in tasks, such as word similarity calculation, entity extraction, and relation extraction. Analogously, the CV algorithms operate on vector representation of images. *Graph embedding* is a generalization of *word embedding*, but for graph-structured input (Hamilton 2020).

Algorithms using vector representations have excelled at many tasks, for example, web search and image recognition. Using web search of today, we can answer questions such as: Who was the prime minister of the UK in October of 1956? But the search fails if the question is modified to an unusual combination of inference steps, for example, Who was the prime minister of the UK when Theresa May was born? Humans have little difficulty in understanding such questions (Lenat 2019a; Lenat 2019b). The limitations of vector representations can be addressed 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 inference and reasoning. The use of graph embeddings with a neural network—also known as *machine learning with graphs*—is 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 while recognizing 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 (plant) has part B (leaf), and B has property P (minty) then A has property P; (b) If A (person) likes property P (minty leaves) of B (plant), then A likes B; and (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 versus machine curation

Industrial KGs, such as 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 one could create such KGs exclusively through automated methods (also referred to as machine curation) versus creation through human effort. This tradeoff is illustrated via two examples based on the MAG and APG, which 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 used machine curation to solve the problem of uniquely identifying authors and their publications (Wang et.al 2020). 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. This approach relies on the authors and publishing organizations contributing manual effort to annotating documents with DOIs and ORCIDs. However, human curation of even such simple tasks has been problematic for several reasons. First, such identifiers have had low human readability discouraging their use. 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 by identifying a publication by its contents and disambiguating authors based on their field(s) of research, affiliation(s), coauthor(s), and other factors that are more natural to humans.

The APG is multilingual and aims to collect product knowledge for millions of categories of products and thousands of attributes of each of those products. While one might reasonably assume that vendors interested in selling their products via Amazon might volunteer structured information that could be directly input into the APG, that is not the case in practice, and the structured data are sparse and noisy. However, creating the APG entirely through human curation would have required hundreds of person years of effort. Machine curation techniques were leveraged at different levels of scaling. To get the project off the ground, highly accurate automated knowledge extraction models were created to generate trustworthy data on a small scope of products, where each model extracted knowledge for a single attribute from a single product domain (Zheng et al. 2018). Even though neural networks were explored to automate the process, tremendous manual work was involved to create training data, conduct human evaluation, and to identify postprocessing rules to remove extraction noise. The next level of scaling aimed to reduce modeling cost through AutoML and automatic cleaning techniques (Wang et al. 2020) so that manual tuning for each knowledge extraction model could be significantly reduced. Scaling further required reducing the total number of models required for the variety of knowledge to be extracted, which was achieved through transfer learning techniques such that a model can extract knowledge for multiple attributes and for multiple domains (Karamanolakis, Ma, and Dong 2020). The final level of scaling aimed to increase knowledge extraction yield through 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 required leveraging techniques such as named entity recognition, closed information extraction, knowledge cleaning, and knowledge-based question answering.

The Wikidata KG was launched to address the problem that data in Wikipedia is buried across 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 data is inconsistent—the population numbers in these different Wikipedia documents are all different. Having been founded on the principle of plurality, it is not easy, or even possible, to arrive at a global consensus 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 in values. Checking, verifying, and allowing such a plurality of data is something the Wikipedia community has been doing for years. Wikidata's human curation effort involves a community of over 400,000 editors, with over 20,000 active editors. In this process, 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 a list of standard identifiers for items that appear in its corpus, which are now increasingly being used in commercial KGs.

Finally, consider Cyc, the largest available knowledge base that captures complex human common sense. The Cyc knowledge base was largely created through human curation because the project aims to capture "hidden" knowledge that is not explicitly written down in text and, thus, cannot be automatically extracted. Early versions of Cyc employed representations like present-day KGs. Since 1989, Cyc has used a representation language called CycL which is based in higher-order logic and nested

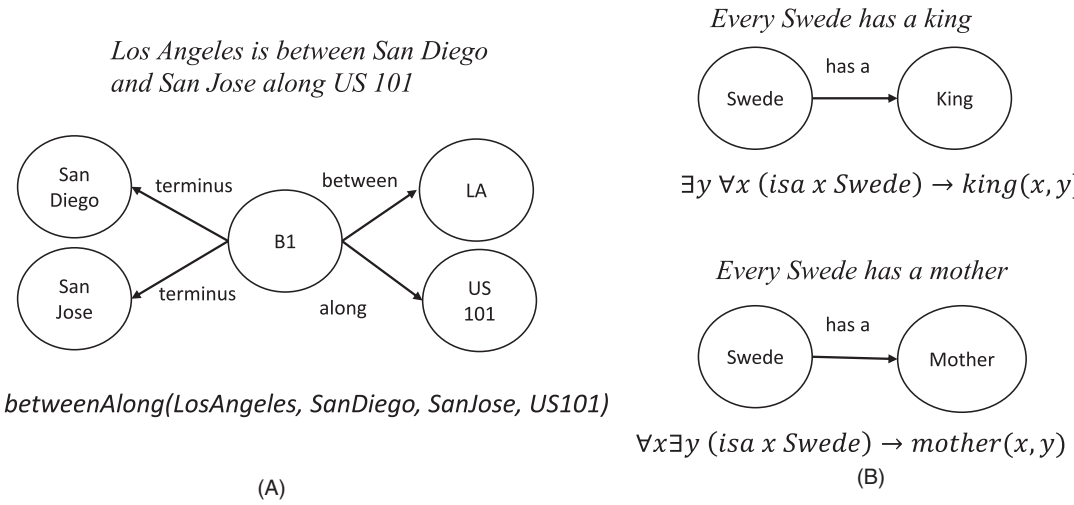

FIGURE 5 Example sentences and their representation in knowledge graph and first order logic

modals (Lenat and Guha 1991). CycL was needed to represent and 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 present NLP techniques even if the knowledge to be entered had been explicitly written down. Cyc is building increasingly automatic tools that help lower the bar for creation and modification of its KB. The project's Knowledge Axiomatization Institute (KNAXI) is also interested in education and professional training in "ontological engineering" at all education levels to facilitate creation of CycL knowledge bases.

Little semantics versus big semantics

The *big semantics* perspective may be viewed as one that advocates for capturing more meaning about concepts. Whereas, the *little semantics* perspective, is focused on capturing/recording the basic facts and not so much the concept meanings. A KG defined as a directed labeled graph is a representative technique of the little semantics approach. The representation kanguage CycL is a representative technique of the big semantics approach.

Using only directed labeled graph representation for KGs has its inherent limitations. A simple example of such a limitation is in representing the statement: *Los Angeles is between San Diego and San Jose along US 101*. This statement could be captured in a directed labeled graph using a technique known as *reification* but requires multiple triples (see Figure 5A). The statement can be captured directly if we allow four-place predicates which are not supported in directed graphs—although many implementations of graph and semantic web databases do include this capability. For this example, the KG representation is akin to using assembly language as opposed to a higher-level programming language. Use of triples and reification makes downstream tasks such as natural language generation more difficult as they must now 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 meanings (see Figure 5B). It is possible to extend the directed graphs in a variety of ways to correctly capture the semantics of the example considered in Figure 5B (Chaudhri et al. 2004; Sowa 2008), but such extensions lose the simplicity offered by the triple representation. Not surprisingly, similar efforts are underway for machine learning of nonbinary relationships as well (Fatemi et al. 2019).

Despite the above stated limitations of the directed labeled graph representation for KGs, it has been found useful for solving many practical problems that are well served by *little semantics*. Wikidata, Data Commons, MAG, and APG all employ 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?—which is precisely the question being addressed by multiple OKN 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

TABLE 2 Difference between research on semantic networks and knowledge graphs

	Semantic networks	Knowledge graphs
Scale	Thousands of objects	Billions of objects
	Complex logical inference	Scalable graph algorithms
		Neuro symbolic reasoning
Development methods	Top-down design	Bottom-up design
	Complex rules and ontologies	Triples and embeddings
Modes of construction	Knowledge engineering	Knowledge engineering, crowdsourcing, machine learning

support codesignation of objects in different KGs, that is, which objects reference the same real object? Robust solutions to these types of issues will be critical in advancing our ability to create open KGs.

SUMMARY AND CONCLUSION

KGs have emerged as indispensable information structures that enable access, integration, and use of the vast amounts of data that are currently being generated. A KG also serves the purpose of capturing knowledge learned and used by modern machine learning methods. The most notable uses of directed labeled graphs in AI and databases (data modeling) have taken the form of data graphs, taxonomies, and ontologies. While this representation schema may fall short of the full capability of reasoning and inferencing that is required by general-purpose repositories of knowledge for AI programs, it still provides a scalable and powerful representation that serves many needs.

Even though a directed labeled graph is a common thread linking present day KGs with the early semantic networks in AI, there are some important differences in the research methodology and technical problems addressed. Early semantic networks were created by top-down design methods and manual knowledge engineering processes. They never reached the size and scale of today's KGs. In contrast, modern KGs tend to be large in scale; employ bottom-up development techniques; and employ manual as well as automated strategies for their construction. The differences are summarized in Table 2.

The emphasis in the early AI semantic networks was on complex logical inferencing, in contrast to the focus on supporting analytics operations in modern KGs. Furthermore, vast proliferation of available data, difficulty in arriving at a top-down schema design for data integration, and the data-driven nature of machine learning have all led to a bottom-up methodology for creating KGs. Contemporary KGs are also supplementing manual knowledge engineering techniques with crowdsourcing and significant automation that is now possible through progress in machine learning. However, we posit that modern KG construction methods should also learn the lessons from classical knowledge representation, as there is much to benefit from the substantial body of prior research without reinventing available methods and tools.

We conclude by noting that making progress does not require us to settle all the debates, for example, on symbolic representation versus vector representation, manual curation versus machine curation, and little semantics versus big semantics. Indeed, as reflected by the ethos of the NSF Convergence Accelerator program, we should drive future research by exploring and prototyping various approaches in the context of real-world use cases. Setting a use-inspired context enables us to justify the need and helps specify the requirements for the specific innovations for KGs to have the maximum societal and scientific impact.

ACKNOWLEDGMENTS

This work has been partially supported by National Science Foundation's Convergence Accelerator program. We sincerely thank Dr. RV Guha for his contributions and insightful comments on the paper.

CONFLICT OF INTEREST

No conflict of interest has been declared by the author(s).

ENDNOTES

[i] https://web.stanford.edu/class/cs520/
[ii] https://www.gqlstandards.org/
[iii] https://blog.google/products/search/introducing-knowledge-graph-things-not/
[iv] https://en.wikipedia.org/wiki/Winterthur
[v] https://en.wikipedia.org/wiki/Ontario,_California
[vi] https://www.wikidata.org/wiki/Q9125
[vii] https://www.wikidata.org/wiki/Q488134
[viii] https://id.loc.gov/authorities/names/n50013808.html
[ix] http://datacommons.org
[x] http://factset.com
[xi] https://nlp.stanford.edu/projects/glove/

REFERENCES

Baranzini, S., K. Börner, J. Morris, C. A. Nelson, K. Soman, E. Schleimer, M. Keiser, M. Musen, R. Pearce, T. Reza, B. Smith, B. Herr, B. Oskotsky, A. Rizk-Jackson, K. Rankin, S. Sanders, R.

Bove, P. Rose, S. Israni, and S. Huang 2022. "A Biomedical Open Knowledge Network Harnesses the Power of AI to Understand Deep Human Biology." *AI Magazine* 43(1): 46–58.

Berners-Lee, T., J. Hendler, and O. Lassila 2001. "The semantic web." *Scientific American* 284(5): 34–43.

Borgida, A., and J. Mylopoulos 2009. "A Sophisticate's Guide to Informational Modeling." In *Metamodeling for Method Engineering*. Cambridge, MA MIT: MIT Press.

Brachman, R. J., and H. J. Levesque 1984. "The Tractability of Subsumption in Frame-Based Description Languages." *In the proceedings of the annual conference of the Association for the Advancement of Artificial Intelligence* 84: 34–7.

Brin, S., and L. Page 1999. "The anatomy of a large-scale hypertextual web search engine." *Computer Networks and ISDN Systems* 30(1-7): 107–17.

Buneman, P. 1997. "Semistructured Data." In *Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* 117–21.

Cafarella, M., M. Anderson, I. Beltagy, A. Cattan, S. Chasins, I. Dagan, D. Downey, O. Etzioni, S. Feldman, T. Gao, T. Hope, K. Huang, S. Johnson, D. King, K. Lo, Y. Lou, M. Shapiro, D. Shen, S. Subramanian, L. Wang, Y. Wang, Y. Wang, D. Weld, J. Vo-Phamhi, A. Zeng, and J. Zou 2022. "Infrastructure for Rapid Open Knowledge Network Development." *AI Magazine* 43(1): 59–68.

Chaitan, B., Scott, B., Lara, C., Aurali, D., Pradeep, F., Alex, L., Douglas, M., Ibrahim, M., Linda, M., Michael, P., Michael, R., Shelby, S., Nicole, T. The NSF Convergence Accelerator Program, AI Magazine 43(1).

Chaudhri, V., K. Murray, J. Pacheco, P. Clark, B. Porter, and P. Hayes 2004. "Graph-Based Acquisition of Expressive Knowledge." In International Conference on Knowledge Engineering and Knowledge Management, 231–47. Berlin, Heidelberg: Springer.

Chen, V. S., P. Varma, R. Krishna, M. Bernstein, C. Re, and L. Fei-Fei 2019. "Scene Graph Prediction with Limited Labels." In *Proceedings of the IEEE/CVF International Conference on Computer Vision* 2580–90.

Codd, E. F. 1982. "Relational Database: A Practical Foundation for Productivity." In *Communications of the ACM* 25(2): 109–17.

Cygniak, R., D. Wood, and M. Lanthaler 2014. "RDF 1.1 Concepts and Abstract Syntax. https://www.w3.org/TR/rdf11-concepts/

Ding, W., V. K. Chaudhri, N. Chittar, and K. Konakanchi May 2021. "JEL: Applying End-to-End Neural Entity Linking in JPMorgan Chase." In *Proceedings of the AAAI Conference on Artificial Intelligence* 35(17): 15301–8.

Dunietz, J., G. Burnham, A. Bharadwaj, O. Rambow, J. Chu-Carroll, and D. Ferrucci 2020. "To Test Machine Comprehension, Start by Defining Comprehension." *arXiv preprint arXiv:2005.01525 [cs.CL]*.

Fatemi, B., P. Taslakian, D. Vazquez, and D. Poole 2019. "Knowledge Hypergraphs: Extending Knowledge Graphs Beyond Binary Relations." *arXiv preprint arXiv:1906.00137*.

Feigenbaum, E. A. 1984. "Knowledge Engineering." *Annals of the New York Academy of Sciences* 426(1): 91–107.

Ferrucci, D., E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, and N. Schlaefer 2010. "Building Watson: An overview of the DeepQA project." *AI Magazine* 31(3): 59–79.

Guha, R. V. 1996. *Meta-Content Format*. Working Paper, Apple Computers.

Guha, R. V., D. Brickley, and S. Macbeth 2016. "Schema. org: Evolution of Structured Data on the Web." *Communications of the ACM* 59(2): 44–51.

Gutiérrez, G., and J. F. Sequeda 2021. "Knowledge Graphs." *Communications of the ACM* 64(3): 96–104.

Hamilton, W. L. 2020. "Graph Representation Learning." *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14(3): 1–159.

Hayes, P. J. 1981. "The Logic of Frames." In *Readings in Artificial Intelligence*, 451–8. San Mateo, CA: Morgan Kaufmann.

Hendler, J. 2007. "The Dark Side of the Semantic Web." *IEEE Intelligent Systems* 22(1): 2–4.

Janowicz, K., P. Hitzler, W. Li, D. Rehberger, M. Schildhauer, R. Zhu, C. Shimizu, C. Fisher, L. Cai, G. Mai, J. Zalewski, Z. Lu, S. Stephen, S. Gonzalez, A. Carr, A. Schroeder, D. Smith, L. Usery, D. Varanka, D. Wright, S. Wang, Y. Tian, Z. Liu, and Z. Gu 2022. "Know, Know Where, KnowWhereGraph: A Densely Connected, Cross-Domain Knowledge Graph and Geo-Enrichment Service Stack for Applications in Environmental Intelligence." *AI Magazine* 43(1): 30–39.

Johnson, J., T. Narock, J. Singh-Mohudpur, D. Fils, K. Clarke, S. Saksena, A. Shepherd, S. Arumugam, and L. Yeghiazarian 2022. "Knowledge Graphs to Support Real-Time Flood Impact Evaluation." *AI Magazine* 43(1): 40–45.

Kalyanpur, A., T. Breloff, D. Ferrucci, A. Lally, and J. Jantos 2020. "Braid: Weaving Symbolic and Neural Knowledge into Coherent Logical Explanations." *arXiv preprint arXiv:2011.13354*.

Karamanolakis, G., J. Ma, and X. L. Dong 2020. "Txtract: Taxonomy-Aware Knowledge Extraction for Thousands of Product Categories." *arXiv preprint arXiv:2004.13852*.

Kowalski, R. 2014. "History of Logic Programming." *Computational Logic* 9: 523–69.

Lenat, D. B., and R. V. Guha 1991. "The Evolution of CycL, the Cyc Representation Language." *ACM SIGART Bulletin* 2(3): 84–7.

Lenat, D. B. 1995. "CYC: A Large-Scale Investment in Knowledge Infrastructure." *Communications of the ACM* 38(11): 33–8.

Lenat, D. B. 2019a. "What AI Can Learn From Romeo & Juliet, Forbes," July 3, 2019. https://www.forbes.com/sites/cognitiveworld/2019/07/03/what-ai-can-learn-from-romeo--juliet/?sh=7f96d5851bd0

Lenat, D. B. 2019b. "Not As Good As Gold: Today's AI are Dangerously Lacking in AU (Artificial Understanding), Forbes," February 18, 2019. https://www.forbes.com/sites/cognitiveworld/2019/02/18/not-good-as-gold-todays-ais-are-dangerously-lacking-in-au-artificial-understanding/?sh=b84ea9536dd4

Lin, R., X. He, J. Feng, N. Zalmout, Y. Liang, L. Xiong, and X. L. Dong 2021. "PAM: Understanding Product Images in Cross Product Category Attribute Extraction." *arXiv preprint arXiv:2106.04630*.

McCarthy, J. 1989. "Artificial Intelligence, Logic and Formalizing Common Sense." In *Klüver Academic*. http://jmc.stanford.edu/articles/ailogic.html

McComb, D. 2018. *Software Wasteland: How the Application-Centric Mindset is Hobbling our Enterprises*. Basking Ridge, NJ, USA: Technics Publications.

McComb, D. 2019. *The Data-Centric Revolution: Restoring Sanity to Enterprise Information Systems*. Basking Ridge, NJ, USA: Technics Publications.

Mikolov, T., K. Chen, G. Corrado, and J. Dean 2013. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*.

Mostafazadeh, N., A. Kalyanpur, L. Moon, D. Buchanan, L. Berkowitz, O. Biran, and J. Chu-Carroll 2020. "Glucose: Generalized and contextualized story explanations." *arXiv preprint arXiv:2009.07758*.

Newell, A. 1982. "The Knowledge Level." *Artificial Intelligence* 18(1): 87–127.

Noy, N., Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor 2019. "Industry-Scale Knowledge Graphs: Lessons and Challenges." *Communications of the ACM* 62(8): 36–43.

Pah, A., D. Schwartz, S. Sanga, C. Alexander, K. Hammond, and L. Amaral 2022. "The Promise of AI in an Open Justice System." *AI Magazine* 43(1): 69–74.

Peng, P., L. Zou, M. T. Özsu, and D. Zhao 2018. "Multi-Query Optimization in Federated RDF Systems." In *International Conference on Database Systems for Advanced Applications*, 745–65. Cham: Springer.

Pérez, J., M. Arenas, and C. Gutierrez 2009. "Semantics and Complexity of SPARQL." *ACM Transactions on Database Systems (TODS)* 34(3): 1–45.

Robinson, I., J. Webber, and E. Eifrem 2015. "*Graph Databases: New Opportunities for Connected Data*. North Sebastopol, CA: O'Reilly Media, Inc.

Sequeda, J. F., and O. Lassila 2021. *Designing and Building Enterprise Knowledge Graphs*. Williston, VT: Morgan and Claypool Publishers.

Sap, M., R. Le Bras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N. A. Smith, and Y. Choi 2019. "Atomic: An Atlas of Machine Commonsense for If-Then Reasoning." In *Proceedings of the AAAI Conference on Artificial Intelligence* 33(1): 3027–35.

Sowa, J. F. 2008. "Conceptual graphs." *Foundations of Artificial Intelligence* 3: 213–37.

Taylor, R. W., and R. L. Frank 1976. "CODASYL Data-Base Management Systems." *ACM Computing Surveys (CSUR)* 8(1): 67–103.

Vrandečić, D., and M. Krötzsch 2014. "Wikidata: A Free Collaborative Knowledgebase." *Communications of the ACM* 57(10): 78–85.

Waagmeester, A., E. L. Willighagen, A. I. Su, M. Kutmon, J. E. L. Gayo, D. Fernández-Álvarez, Q. Groom, P. J. Schaap, L. M. Verhagen, and J. J. Koehorst 2021. "A Protocol for Adding Knowledge to Wikidata: Aligning Resources on Human Coronaviruses." *BMC Biology* 19(1): 1–14.

Wang, K., Zhihong, S., Chiyuan, H., Chieh-Han, W., Yuxiao, D., and Anshul, K. "Microsoft academic graph: When experts are not enough." *Quantitative Science Studies 1*, no. 1 (2020): 396–413.

Wang, Y., Y. E. Xu, X. Li, X. L. Dong, and J. Gao 2020. "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* 2533–41.

Woods, W. A. 1975. "What's in a Link: Foundations for Semantic Networks." In *Representation and Understanding*, 35–82. San Mateo, CA: Morgan Kaufmann.

Wu, F., R. Hoffmann, D. S. Weld 2008. "Information Extraction from Wikipedia: Moving Down the Long Tail." In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 731–9.

Yan, J., N. Zalmout, Y. Liang, C. Grant, X. Ren, and X. L. Dong 2021. "AdaTag: Multi-Attribute Value Extraction from Product Profiles with Adaptive Decoding." *arXiv preprint arXiv:2106.02318*.

Zheng, G., S. Mukherjee, X. L. Dong, and F. Li 2018. "Opentag: Open Attribute Value Extraction from Product Profiles." In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 1049–58.

Zhu, Y., O. Groth, M. Bernstein, and L. Fei-Fei 2016. "Visual7w: Grounded Question Answering in Images." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 4995–5004.

AUTHOR BIOGRAPHIES

Vinay K. Chaudhri is currently an Executive Director at JPMorgan Chase & Co. working on AI for the financial services Industry. He was a visiting lecturer In the Department of Computer Science at Stanford University when this work was performed. Prior to that he was at SRI International where he worked on Project Halo that created an Intelligent Textbook, and on Project Calo that was spun off as SIRI and was later acquired by Apple.

Chaitanya Baru is Distinguished Scientist at San Diego Supercomputer Center, UC San Diego. From 2014 to 2018, he was Senior Advisor for Data Science at the National Science Foundation where he provided leadership for data programs including BIGDATA, Big Data Hubs, TRIPODS, Data Science Corps. From 2019 to 2021, he was Senior Advisor for the Convergence Accelerator and a member of the team that established the program.

Naren Chittar is a Managing Director and the head of AI Core Services at JPMorgan Chase. He founded Minhash, an NLP startup that was acquired by Salesforce in 2015. Prior to that he co-created e-Bay's first large scale similar items recommendation engine and image search technology.

Xin Luna Dong is the Head Scientist at Meta AR/VR Assistant. Prior to this, she was Sr. Principal Scientist at Amazon, leading the efforts to build Amazon Product Knowledge Graph. She has co-authored books "Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases" and "Big Data Integration", was awarded ACM Distinguished Member and VLDB Early Career Research Contribution Award, and is a PC co-chair for KDD 2022, WSDM 2022, and VLDB 2021.

Michael Genesereth is a professor in the Computer Science Department at Stanford University and a professor by courtesy in the Stanford Law School. He is most known for his work on Computational Logic and

applications of that work in Enterprise Management, Computational Law, and General Game Playing.

James Hendler is the Director of the Institute for Data Exploration and Applications and the Tetherless World Professor of Computer, Web and Cognitive Sciences at Renssalear Polytechnic Institute (RPI). He also is acting director of the RPI-IBM Artificial Intelligence Research Collaboration and serves as a Chair of the Board of the UK's charitable Web Science Trust. Hendler has authored over 400 books, technical papers and articles in the areas of Semantic Web, artificial intelligence, agent-based computing and high-performance processing.

Aditya Kalyanpur is the director of machine learning and natural language processing at Elemental Cognition. Previously, he was one of the key developers of the IBM Watson Question Answering system that won the Jeopardy! challenge in 2011, and a recipient of the AAAI Feigenbaum Prize for this work.

Douglas B. Lenat is one of the world's leading computer scientists, founder of the Cyc project and Cycorp. Dr. Lenat has been a Professor of Computer Science at Carnegie-Mellon University and Stanford University, was awarded the biennial IJCAI Computers and Thought Award, which is the highest honor in artificial intelligenceHe was the first Fellow of the Association for the Advancement of Artificial Intelligence (AAAI); is a Fellow of the American Academy for the Advancement of Science (AAAS) and the Cognitive Science Society, and the only person to have served on the scientific advisory boards of both Microsoft and Apple.

Juan Sequeda is the Principal Scientist at data.world. He joined through the acquisition of Capsenta, a company he founded as a spin-off from his PhD research in Computer Science from The University of Texas at Austin. His goal is to reliably create knowledge from inscrutable data.

Denny Vrandečić works on Abstract Wikipedia at Wikimedia Foundation. Formerly, he was an Ontologist at Google, a Trustees of the Board of Wikimedia Foundation, Wikidata founder, co-developer of Semantic MediaWiki, and an author of several RPG modules for "Das Schwarze Auge".

Kuansan Wang is a partner architect at Microsoft Search. Formerly, he was the managing director of Microsoft Research (MSR) Outreach, responsible for Microsoft Academic Services (MAS).

How to cite this article: Chaudhri, V. K., C. Baru, N. Chittar, X. L. Dong, M. Genesereth, J. Hendler, A. Kalyanpur, D. Lenat, J. Sequeda, D. Vrandečić, and K. Wang 2022. "Knowledge graphs: Introduction, history, and perspectives." *AI Magazine* 43: 17–29. https://doi.org/10.1002/aaai.12033