EDITOR: Amit Sheth, amit@sc.edu

DEPARTMENT: KNOWLEDGE GRAPH

The EMPWR Platform: Data and Knowledge-Driven Processes for the Knowledge Graph Lifecycle

Hong Yung Yip and Amit Sheth, *University of South Carolina, Columbia, SC, 29208, USA*

The unparalleled volume of data generated has heightened the need for approaches that can consume these data in a scalable and automated fashion. Although modern data-driven, deep-learning-based systems are cost-efficient and can learn complex patterns, they are black boxes in nature, and the underlying input data highly dictate their world model. Knowledge graphs (KGs), as one such technology, have surfaced as a compelling approach for using structured knowledge representation to support the integration of knowledge from diverse sources and formats. We present Empower (EMPWR), a comprehensive KG development and lifecycle support platform that uses a broad variety of techniques from symbolic and modern data-driven systems. We discuss the sets of system design guiding principles used to develop EMPWR, its system architectures, and workflow components. We illustrate some of EMPWR's abilities by describing a process of creating and maintaining a KG for the pharmaceuticals domain.

With the rapid advancement and widespread use of digital technologies, we are witnessing unprecedented data generated, from social media interactions to online transactions, and from sensor readings to health-care records. The unparalleled volume, variety, and velocity of data being generated in the current digital era have rendered the manual rule-based declarative approach to symbolic knowledge acquisition, representation, and reasoning less effective and has thus propelled the need for approaches that can *consume* (process, analyze, and glean insights on) these data in a scalable and automated fashion. As such, the emergence of modern data-driven systems and the continuous evolution of Artificial Intelligence (AI) applications can be seen as clear attributions and exemplars of how the abundance of data has transformed the way we live, work, and interact with the world around us as well as support the efficient functioning of modern society.

These systems based on neural networks and deep learning (e.g., ChatGPT) are cost-efficient for use (albeit expensive to build) and can consume, recognize, and learn complex patterns and relationships in the underlying data on their own, without the need for arduous human labor in knowledge curation and feature engineering. In addition, modern self-supervised systems (e.g., zero-shot learning) can learn from a small pool of data without needing a large investment of human annotations, and appeal to individuals and organizations with resource constraints. The ease of scalability and deployment of current state-of-the-art architectures is the cherry on top. However, they are not silver bullets as their *world model*; the spatial and temporal representations and understanding of the environment are highly dictated by the underlying input data. In other words, unsanitized data and nonvalidated inputs may lead to factually incorrect models, biases, and hallucinations, which can be adversarial. In addition, these systems are usually black box in nature and fall short in explainability and provenance as their performance and capabilities are determined by tuning their underlying weights and parameters, which need to be more human-understandable. Other aspects, such as ethics, governance, and safety, are still at the forefront of research. Therefore, we should recognize the merit of

1089-7801 © 2024 IEEE
Digital Object Identifier 10.1109/MIC.2023.3339858
Date of current version 30 January 2024.

traditional symbolic approaches and understand the underlying sources, characteristics, and implications of these data, models', and systems' deluges to harness their potential and navigate the complexities they bring.

Traditional symbolic technologies (e.g., the Semantic Web) rely on various representational and logic-based formalisms to provide a foundation for building knowledge representation and reasoning systems. They provide notational efficacy and declarative capabilities that can be used to make implicit data explicit, enabling high-quality linguistic and situational knowledge and the ability to formally capture the structure and behavior of the objects around us and audit the reasoning. These systems are more computationally tractable, and domain scope and constraints can be easily enforced, which, in turn, supports data governance and provenance and provides a more explainable output. Despite the advantages, these systems are being shied away from due to the cost of manual labor in knowledge acquisition and curation and the computational complexity, scalability, and brittleness of the unrepresented information compared to the more modern data-driven systems. Nonetheless, as we enter what DARPA describes as the third phase of AI, which involves combining statistical and symbolic approaches (i.e., neurosymbolic AI[7]), the role of *knowledge* is becoming indispensable in making sense of data, and we are witnessing an increased adoption of knowledge graphs (KGs), which is a form of the Semantic Web approach as a key enabler for data-driven solutions that involve intelligent data transformation into insights, actionable information, and decisions as well as making AI systems more transparent and auditable.

At its core, a KG represents real-world entities as *nodes* and the types of relationships among entities as *edges*. It is founded on *ontology commitment*, where the meanings of entities and relationships that domain experts agree upon are explicitly defined and published. As such, it has surfaced as a compelling approach for imparting definitions, structure, and uniformity over raw data and integrating them from diverse sources (typically siloed) and formats (unstructured, semistructured, and structured). Increasingly, KGs are being used to power consumer applications such as search engines [e.g., Google KG (http://bit.ly/google_kg)], social media [e.g., LinkedIn KG (https://bit.ly/linkedingraph)], chatbots, and recommendation systems [e.g., Amazon Product KG (https://bit.ly/amazon-productkg)] as well as health-care research [e.g., Amazon COVID-19 KG (https://bit.ly/amazoncovid19kg)]. However, designing and creating a KG from the ground up requires a substantial up-front investment of time and human labor in knowledge curation. Although tools exist to support the semiautomated development of KGs, they are often (singular) domain and application driven with specific use cases, requirements, and purposes. Most are designed to extract knowledge from the specific corpus. Here, we taxonomize a list of existing KG development tools into several categories.

The following tools specialize in natural language processing (NLP):

- *AI-KG*: an automatically generated KG of AI.[1]
- Automatic KG Creation framework from NLP.[2]

The following tools have been developed for a specific domain and application with target use cases and datasets:

- Learning a Health KG from an electronic medical record.[3]
- Building a PubMed KG.[4]
- *KGen*: a KG generator from biomedical scientific literature.[5]

The following tool has been developed for any domain:

- Heaven Ape (HAPE) (programmable big knowledge graph platform).[6]

In this article, we advocate an approach that hybridizes the multiple techniques from traditional symbolic and modern data-driven systems to design a platform Empower (EMPWR) for the KG lifecycle that encompasses broad-based applications and broad sources of data. We first review the current end-to-end knowledge lifecycle design practice and the associated challenges. We then discuss the sets of system design guiding principles in developing EMPWR, its system architectures, and workflow components. Finally, we illustrate the process of creating a KG with EMPWR, drawing experiences from our work in pharmaceutical KG

DEEP-LEARNING-BASED SYSTEMS

Although deep-learning-based systems are cost-efficient and can learn complex patterns without explicit knowledge curation, they are not silver bullets. The underlying input data highly dictate their world model. On the contrary, knowledge graphs impart definitions, structure, and uniformity.

construction (a partnership with collaborator WIPRO) with more than 6 M triples, 1.5 M nodes, and 3000 relation types and interconnecting knowledge from broad-based open and domain-specific knowledge sources.

THE KG LIFECYCLE

The standard practice of an end-to-end KG lifecycle consists of different phases: 1) design and requirements scoping, 2) data ingestion, 3) data enrichment, 4) storage, 5) consumption, and 6) maintenance. Next, we describe each phase and its associated challenges.

The *design* phase entails scoping the target use case and application's requirements, followed by creating the domain ontology. This is currently one of the more resource-consuming phases due to the involvement of domain experts and communities from different disciplines to congregate on the design and development of the appropriate schemas and representation formats, and assessment of relevant data sources to fit the intended use case. With the sheer availability and ease of accessibility of data today, we believe that a bottom-up approach, where we infer the domain ontology from the underlying data and the ontology is then reviewed and edited by domain experts, can drastically reduce the initial up-front commitment and bootstrap the design process. As such, we aim to streamline, scale, and automate such an approach with EMPWR.

The *data ingestion* phase involves the process of collecting, extracting, and transforming data from various sources [e.g., databases, application programming interfaces (APIs), web scraping, and user entry] and heterogenous formats (e.g., unstructured, semistructured, and structured) into a unified format that can be used to build a KG. This is a fundamental and critical step as it lays the foundation for organizing and connecting data from multiple siloed sources for information discovery and analysis. It requires careful consideration when creating and structuring data pipelines and workflows, mapping *legacy* data to the established ontology, and standardizing the representation. The challenges lie in implementing measures to ensure consistent and reliable transformation, data governance, storage, and access rights (e.g., different groups of users with various levels of access privilege), data validation (e.g., ensuring data sanity), lineage, and provenance (i.e., the ability to pinpoint data origin) as well as efficient scalability to accommodate the volume and diversity of the data. In later sections, we describe how we consider the aforementioned challenges in designing EMPWR.

The *data enrichment* phase involves various processes and methodologies to improve data quality, assign definitions and meanings (e.g., entity class and named relation) to data, expand the initial vocabulary scope with external authoritative knowledge bases, and enforce any constraints per the domain or application specifications. This phase is the most important step as the raw data are contextualized and abstracted into information, which translates into potentially valuable and actionable insights that enable new knowledge discovery. This includes the use of assorted arrays of NLP techniques such as named-entity recognition, relationship extraction, entity mapping and disambiguation, and relationships linking as well as inferencing and reasoning approaches to derive new information, which may be used to expand the initial ontology from the *design* phase.

The *storage* phase is the repository for managing and hosting the KG, typically on graph databases or triple stores for *consumption*. The *consumption* phase typically accommodates the design of user interfaces (e.g., a front-end data portal) and software interfaces (e.g., APIs) to serve both users and developers, respectively, for KG access, management, and queries. In addition, it should support the KG export to various popular formats [e.g., JavaScript Object Notation for Linked Data (JSON-LD), Resource Description Framework (RDF), and transistor-transistor logic (TTL)] to enable import and extension to other graph databases. The *maintenance* phase involves the ongoing effort to maintain the ever-growing schemas and KGs through versioning and suitable provenance measures. The challenge lies in scalability and extensibility, i.e., keeping up with real-world events' dynamicity and temporal updates and evolving graph structures.

DEVELOPING A KNOWLEDGE GRAPH

Developing a knowledge graph (KG) is not a one-off process. It is a lifecycle that consists of different phases and requires continuous maintenance, which necessitates a scalable platform that provides a wide range of capabilities, such as data extraction and ingestion from multiple sources and continuous KG updates with schedulers, with the capacity to scale, both in computation and storage, to ensure timely updates to reflect real-world knowledge changes.

THE EMPWR PLATFORM AND ITS SERVICES

EMPWR is designed to support a comprehensive suite of services, including creating and managing large knowledge graphs (KGs) and enriching, aligning, and mapping with existing KGs. Compared to existing alternatives (Table S1) such as Protege (appropriate for ontology design but without support for populating ontologies with extensive knowledge) or Amazon Comprehend Medical (appropriate for creating KGs in the medical domain from unstructured data only through natural language processing), EMPWR distinguishes itself in terms of the types of knowledge sources supported (unstructured, semistructured, and structured) and its support for interconnecting the KGs that span diverse domains. Its architecture comprises three main modularized workflow components (to support scalability and extensibility): 1) front end, where users interact with the data portal for data upload (ingestion) and knowledge query; 2) knowledge extraction; and 3) knowledge enrichment, with extension to the Common Metadata Framework (https://github.com/HewlettPackard/cmf) by Hewlett Packard Enterprise to capture all the workflow metadata related to the end-to-end KG lifecycle, provenance, and lineage to enable reproducibility and traceability, and the Intelligent Data Store (https://bit.ly/hpe_ids) for scalable compute and storage. We illustrate the steps in creating a pharmaceutical KG, drawing upon experiences from our partnership with collaborator WIPRO with more than 6 M triples, 1.5 M nodes, and 3000 relation types and interconnecting knowledge from broad-based open (i.e., DBpedia, WikiData, and ConceptNet) and domain-specific knowledge sources

Table S1. Distinctions with popular KG tools.

Parameter	EMPWR+Common Metadata Framework	Protege	Amazon Comprehend Medical
Functionality	Comprehensive suite designed to build knowledge graphs (KGs) from unstructured data.	Facilitates manual construction of complex ontologies, enabling deep, domain-specific modeling without automatic extraction.	NLP service designed to extract medical information from unstructured text.
Data handling	Handles unstructured data using automatic knowledge extraction.	Requires structured data in the form of ontologies.	Handles unstructured text, specifically focused on medical documents.
Knowledge extraction and enrichment	Automatic extraction and enrichment using tool kits and large language models with data from external sources.	Does not directly handle knowledge extraction or enrichment from unstructured data.	Specializes in extracting medical information from unstructured text but lacks direct knowledge enrichment.
Best model performance logging and version controlling	Integrates with the Common Metadata Framework to capture metadata, provenance, and lineage.	Does not include this functionality out of the box.	Does not inherently track workflow metadata; this functionality must be added separately.
Schema generation and validation	Automatic schema generation from extracted triples, validated by user input.	It supports user-defined schemas (ontologies) but does not automatically generate or validate these.	Does not provide schema generation or validation; its focus is primarily on medical entity extraction.
Data storage and export	Supports storing in the Intelligent Data Store and exporting in Resource Description Framework OWL, JSON-LD, XML, and TTL formats.	Supports storing and exporting ontologies in various formats.	Does not inherently store or export data; it primarily processes and returns analysis of the input text
User interaction	Includes a front end for user interaction to upload data and query the KG.	Provides a GUI for creating and editing ontologies.	Utilized as an application programming interface but does not have a user interface for direct user interaction.
Domain focus	General tool, not limited to a specific domain.	General tool, not limited to a specific domain.	Specifically focused on the medical domain.
OKN support	Supports creation and editing of ontologies, essential for building and contributing to OKNs.	Despite facilitating ontology creation, lacks direct support for integrating or querying OKNs.	Specialized medical information extractor, not directly oriented toward OKN support.

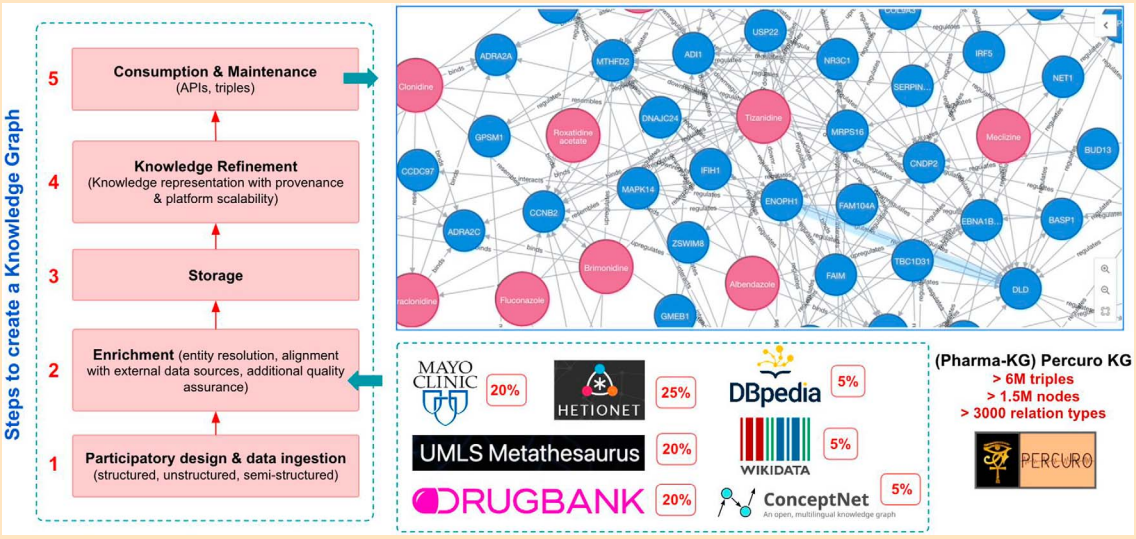

FIGURE S1. Pharma KG (developed by the Artificial Intelligence Institute of the University of South Carolina in collaboration with industry collaborator WIPRO, with requirements/scope driven by major pharma companies). The bottom right shows diverse medical and open knowledge sources from which knowledge was extracted, aligned, merged, enriched, curated, and evaluated against user/application requirements with the help of EMPWR.

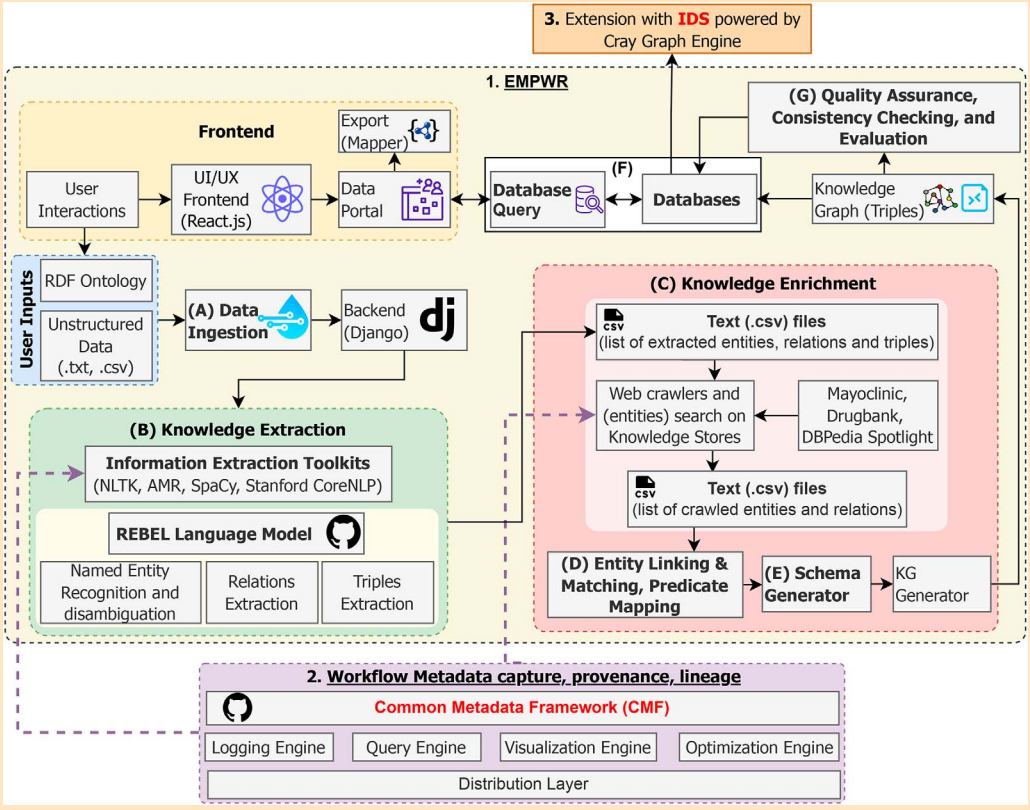

FIGURE S2. Integrating EMPWR, the Common Metadata Framework (CMF), and the Intelligent Data Store (IDS) for the KG's lifecycle. (A) Data ingestion: users may upload a list of unstructured datasets (i.e., MashQA or MEDIQA) via the

[MayoClinic, Hetionet, Unified Medical Language System (UMLS), and Drugbank]. Figure S1 uses the framework illustrated in Figure S2. Many capabilities of EMPWR are illustrated in the demo at http://bit.ly/empwr.

Figure S2. (*Continued*). front-end portal. The ingestion endpoints are extended to incorporate existing KGs stored in popular graph databases (Neo4j, Stardog, and Amazon Neptune) support the ability to curate KGs as well as workflow (B) and (C). (B) Knowledge extraction: the knowledge-extraction module consists of state-of-the-art information-extraction tool kits [Natural Language Toolkit (NLTK), Abstract Meaning Representation (AMR), Advanced Natural Language Processing in Python and Cython (SpaCy), and Stanford CoreNLP] as well as large language models that perform a series of natural language tasks on the underlying datasets, drawing upon our large body of work in entity extraction, compound entity extraction, implicit entity extraction, entity linking, relationship extraction, semantic path computation and ranking, and federated learning. The module is extended with the Knowledge Graph Toolkit (https://github.com/usc-isi-i2/kgtk) from the University of Southern California, Information Sciences Institute (USC ISI) for any KG transformation and manipulation functionalities. (C) Knowledge enrichment: the extracted list of entities and relations are then augmented and enriched with high-quality knowledge from external knowledge stores (DBPedia, MayoClinic, and Drugbank) through our web-crawling engines and publicly available application programming interfaces. For example, an entity paracetamol extracted in (B) is queried through DBPedia Spotlight to retrieve information such as alternative names (e.g., Tylenol and Panadol), synonyms (e.g., N-acetyl-para-aminophenol), and existing Uniform Resource Identifiers (URIs) linked to other knowledge sources (e.g., dbo:pubchem-ID:1983) that are otherwise not available in the underlying datasets ingested in (A). (D) Knowledge alignment: the entities and relations are disambiguated, deduplicated, and mapped using concept similarity and alignment techniques (supervised: synonyms and synsets matching; unsupervised: fuzzy matching, neural networks, large language models, reinforcement learning, and unsupervised learning) as well as our history of work in ontology alignment [Semantic Web Science Association (SWSA)/International Semantic Web Conference (ISWC) 10-year award winning]. Users can validate the aligned KG and connect with community-curated KGs [e.g., WikiData, RxNORM, Unified Medical Language System (UMLS), Geonames, LNEx, Empathi, and KnowWhereGraph] based on similar approaches. As we align knowledge from various ontologies and sources, we also support modeling of the provenance. (E) Schema inference: we generate schemas from the underlying triple instances. For example, the following schema (drug, relieves, symptom) can be inferred from the following triple instances (paracetamol, relieves, headaches) by methods of entity tagging. The inferred schemas are then subjected to users' validation. The invalid schemas and their underlying triple instances are pruned. The KG construction workflow metadata from knowledge extraction (best-performing NLP models with their configurations) to knowledge enrichment is captured and logged by the Hewlett Packard Enterprise CMF framework for provenance and lineage. (F) Knowledge storage and query: The constructed KG is then stored in the Intelligent Data Store with access rights control for semantic querying and visualization. Although we conform to the World Wide Web Consortium (W3C) Resource Description Framework (RDF) Web Ontology Language (OWL) standard as the default format for triples representation, the KG can be exported to various supporting formats such as JSON-LD, XML, and TTL to support downstream use cases on different open source and commercial graph databases such as Virtuoso, Neo4J, and so on. We support the ability to query the graph via the data portal either by (a) natural language and (b) using the RDF Query Language (SPARQL) endpoint or similar capability. (G) Quality assurance, consistency checking, and evaluation: our evaluation encompasses periodic and/or longitudinal analyses and experiments to improve, assess, and evaluate the platform iteratively continually. UI/UX: user interface/user experience.

DEVELOPING A LARGE-SCALE KNOWLEDGE GRAPH

Developing a large-scale knowledge graph (KG) is a continuous process and requires a platform that provides a wide range of capabilities. EMPWR supports data extraction and ingestion from multiple sources, continuous KG updates with schedulers, integration with the Hewlett Packard Enterprise Common Metadata Framework for metadata logging and provenance, and the capacity to scale in computation and storage with the Intelligent Data Store to ensure timely updates to reflect real-world knowledge changes.

THE EMPWR PLATFORM FOR THE KG LIFECYCLE

We bring forth an approach and a set of system design guiding principles and critical elements outlined in the Open Knowledge Network (OKN) report (https://bit.ly/oknreport) to design a platform (which we named *EMPWR*) for KG creation. The guiding principles include governance, ethics, provenance, scalability and interoperability, sustainability, access rights, and data validation.

Next, we describe the Common Metadata Framework (CMF) and the Intelligent Data Store (IDS) framework from Hewlett Packard Enterprise (HPE).

CMF

The CMF (see Figure 1) is an open source framework developed by HPE to record, query, and visualize lineage, the provenance of input–output artifacts (datasets), parameters, and metrics used in computational workflows in a Git-like fashion. The CMF involves the instrumentation of EMPWR's knowledge extraction and enrichment workflow pipelines with CMF's logging API. It is built on machine learning (ML) metadata and data version control and takes a pipeline-centric approach while incorporating features from experiment-centric frameworks. It automatically records pipeline metadata from different stages in the pipeline and offers fine-grained experiment tracking. The framework adopts a data-first approach; the content hash versions and identifies all artifacts recorded. It enables metadata tracking for each workflow variant for reproducibility, audit trails, and traceability. The CMF's metadata are stored in its relational database. The CMF supports importing and exporting metadata in external formats such as OpenLineage to prevent metadata from being siloed into a particular cloud or datastore and facilitate open standards sharing. The CMF also supports querying APIs and a visualization engine for the captured metadata (lineage graphs to visualize the KG construction process). Any site (including a cloud resource) can be set up as a CMF server to facilitate the hosting/sharing/discovery of workflow metadata (metadata hub).

IDS

Integrating the IDS (see Figure 2) into EMPWR serves as both a back-end server and query endpoint for KGs, which regulates access rights, data governance, and

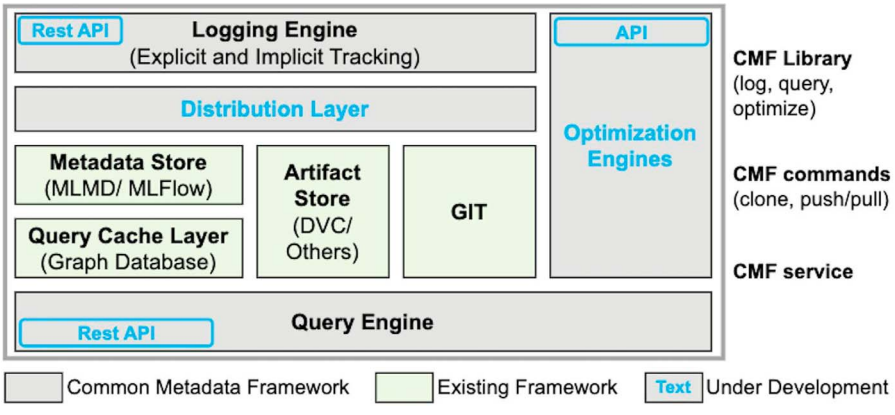

FIGURE 1. CMF architecture. DVC: data version control; MLMD: Machine Learning Metadata; GIT: Global Information Tracker.

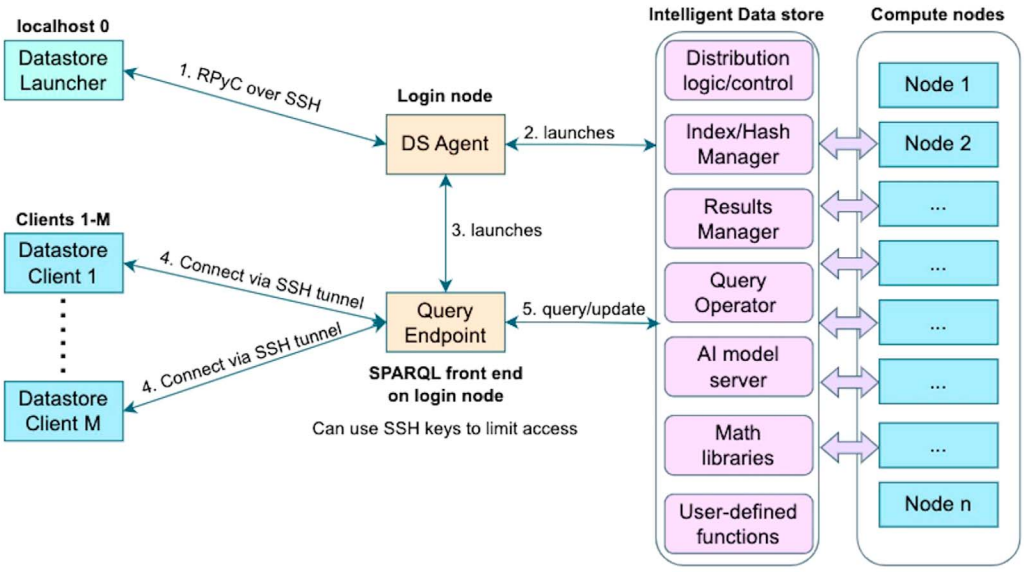

FIGURE 2. A schematic of the IDS. RPyC: Remote Python Call; DS:Data Store; SSH: Secure Shell.

ethical standards. The IDS is an in-memory triple datastore that 1) hosts and serves data in different shapes (documents, graphs, feature vectors, and vector embeddings); 2) allows a pattern search on the hosted data with AI models; 3) supports a query language (e.g., SPARQL) to orchestrate database retrieval (exact search), a pattern search using ML (approximate search), and user-defined functions (domain-specific search); 4) offers easy-to-use programming interfaces for database operations; 5) runs on differentiated server architectures; and 6) is the fastest, massively parallel processing database for unstructured data that scales-out (query latencies in seconds instead of minutes per hours). The core technology behind the IDS is described in Rickett et al.,[8] and the recent success stories hosting drug-discovery KGs are documented in Sukumar et al.[9]

CONCLUSION

In this article, we proposed a hybrid framework that combines the multifaceted approaches from traditional symbolic and modern data-driven systems, as exemplified by the EMPWR platform for KG development. We discussed the advantages and limitations of both families of systems and taxonomized the existing tools for creating KGs. We then reviewed the KG lifecycle design practice and proposed a platform: EMPWR, which supports broad-based applications and broad sources of data for the large-scale development and maintenance of KGs.

The opinions expressed in this article are those of the authors and not the sponsors.

ACKNOWLEDGMENTS

This effort was supported, in part, by WIPRO, and by National Science Foundation Grant 2133842: Early-concept Grants for Exploratory Research (EAGER): Advancing Neuro-Symbolic AI With Deep Knowledge-Infused Learning; and Grant 2335967: EAGER: Knowledge-Guided Neuro-Symbolic AI With Guardrails for Safe Virtual Health Assistants. We also acknowledge the Artificial Intelligence Institute of the University of South Carolina's collaboration with Aalap Tripathy and Sreenivas Rangan Sukumar of HPE on the CMF, and thank Revathy Venkataramanan for the feedback.

REFERENCES

1. D. Dessì, F. Osborne, D. Reforgiato Recupero, D. Buscaldi, E. Motta, and H. Sack, "AI-KG: An automatically generated knowledge graph of artificial intelligence," in *Proc. 19th Int. Semantic Web Conf. Semantic Web II (ISWC)*, Athens, Greece: Springer International Publishing, Nov. 2020, pp. 127–143.
2. N. Kertkeidkachorn and R. Ichise, "An automatic knowledge graph creation framework from natural language text," *IEICE Trans. Inf. Syst.*, vol. 101, no. 1, pp. 90–98, 2018, doi: 10.1587/transinf.2017SWP0006.
3. M. Rotmensch, Y. Halpern, A. Tlimat, S. Horng, and D. Sontag, "Learning a health knowledge graph from electronic medical records," *Scientific Rep.*, vol. 7,

no. 1, 2017, Art. no. 5994, doi: 10.1038/s41598-017-05778-z.

4. J. Xu et al., "Building a PubMed knowledge graph," *Scientific Data*, vol. 7, no. 1, 2020, Art. no. 205, doi: 10.1038/s41597-020-0543-2.
5. A. Rossanez, J. C. Dos Reis, R. D. S. Torres, and H. de Ribaupierre, "KGen: A knowledge graph generator from biomedical scientific literature," *BMC Med. Informat. Decis. Making*, vol. 20, no. 4, pp. 1–24, Dec. 2020, doi: 10.1186/s12911-020-01341-5.
6. L. U. Ruqian et al., "HAPE: A programmable big knowledge graph platform," *Inf. Sci.*, vol. 509, pp. 87–103, Jan. 2020, doi: 10.1016/j.ins.2019.08.051.
7. A. Sheth, K. Roy, and M. Gaur, "Neurosymbolic artificial intelligence (Why, What, and How)," *IEEE Intell. Syst.*, vol. 38, no. 3, pp. 56–62, May/Jun. 2023, doi: 10.1109/MIS.2023.3268724.
8. C. D. Rickett, K. J. Maschhoff, and S. R. Sukumar, "Massively parallel processing database for sequence and graph data structures applied to rapid-response drug repurposing," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2020, pp. 2967–2976, doi: 10.1109/BigData50022.2020.9378331.
9. S. R. Sukumar et al., "The convergence of HPC, AI and Big Data in rapid-response to the COVID-19 pandemic," in *Proc. Smoky Mountains Comput. Sci. Eng. Conf.*, Cham, Switzerland: Springer International Publishing, Oct. 2021, pp. 157–172.

HONG YUNG (JOEY) YIP is a Ph.D. student focusing on semantic web, natural language understanding, and generative AI at the Artificial Intelligence Institute, University of South Carolina, Columbia, SC, 29208, USA. Contact him at hyip@email.sc.edu.

AMIT SHETH is the NCR Chair and Professor of Computer Science & Engineering, University of South Carolina, Columbia, SC, 29208, USA, and the founding director of the Artificial Intelligence Institute, University of South Carolina. Contact him at amit@sc.edu.