

A Community-driven vision for a new Knowledge Resource for AI

Vinay K Chaudhri, Chaitan Baru, Brandon Bennett, Mehul Bhatt, Darion Cassel,
 Anthony G Cohn, Rina Dechter, Esra Erdem, Dave Ferrucci,
 Ken Forbus, Gregory Gelfond, Michael Genesereth, Andrew S. Gordon,
 Benjamin Grosz, Gopal Gupta, Jim Hendler, Sharat Israni,
 Tyler R. Josephson, Patrick Kyllonen, Yuliya Lierler, Vladimir Lifschitz,
 Clifton McFate, Hande Küçük McGinty, Leora Morgenstern,
 Alessandro Oltramari, Praveen Paritosh, Dan Roth, Blake Shepard,
 Cogan Shimizu, Denny Vrandečić, Mark Whiting, Michael Witbrock

July 24, 2025

Abstract

The long-standing goal of creating a comprehensive, multi-purpose knowledge resource, reminiscent of the 1984 Cyc project, still persists in AI. Despite the success of knowledge resources like WordNet, ConceptNet, Wolfram|Alpha and other commercial knowledge graphs, verifiable, general-purpose widely available sources of knowledge remain a critical deficiency in AI infrastructure. Large language models struggle due to knowledge gaps; robotic planning lacks necessary world knowledge; and the detection of factually false information relies heavily on human expertise. What kind of knowledge resource is most needed in AI today? How can modern technology shape its development and evaluation? A recent AAAI workshop gathered over 50 researchers to explore these questions. This paper synthesizes our findings and outlines a community-driven vision for a new knowledge infrastructure. In addition to leveraging contemporary advances in knowledge representation and reasoning, one promising idea is to build an open engineering framework to exploit knowledge modules effectively within the context of practical applications. Such a framework should include sets of conventions and social structures that are adopted by contributors.

1 Introduction

The Cyc project, started in 1984, created the first large-scale database of commonsense knowledge. The initiative continues to this day with its aim to provide a comprehensive ontology and knowledge base of commonsense knowledge to enable human-like reasoning for AI systems. In the concluding paragraph of his Communications of the Association of Computing Machinery (CACM) 1995 article *A Large-Scale Investment in Knowledge Infrastructure* [49], Cyc’s founder Douglas B. Lenat wrote:

Is Cyc necessary? How far would a user get with something simpler than Cyc but that lacks everyday commonsense knowledge? Nobody knows; the question will be settled empirically. Our guess is most of these applications will eventually tap the synergy in a suite of sources (including neural nets and decision theory), one of which will be Cyc.

Although 30 years have passed since the above article was written, AI research community has not conclusively settled [10] the question “How far would a user get with something simpler than Cyc but that lacks everyday commonsense knowledge?” However, it is clear that significant strides have been made in addressing many of the tasks that were original Cyc use cases, including information retrieval, semi-automatically linking multiple heterogeneous external information sources, spelling and grammar correction, machine translation, natural language understanding and speech understanding. Much of this progress has been facilitated by rapid, profound, and synergistic advancements in neural networks, automated agents,

and knowledge graphs [11]. The increasing importance of knowledge graphs is in line with Lenat’s early vision for Cyc, as he presented in 1985 [50].

Knowledge graphs that have had commercial impact, for example, Google’s knowledge graph, have primarily focused on capturing relationships among real-world entities and have sidestepped the challenge of capturing everyday commonsense knowledge. Likewise, even though large language models (LLMs) perform very well on tasks that require knowledge about the world, that knowledge is most often implicit. Furthermore, it has been difficult to make formal guarantees about how and when LLMs leverage this implicit knowledge [66].

The lack of large-scale commonsense knowledge bases hinders practical AI applications. For example, robotic task and motion planning would significantly benefit from world knowledge, and commonsense task descriptions to be performed by a robot [40]. At present, such knowledge is custom built for each project. Pressing problems such as identifying factually false information [57] require trusted knowledge; without it, systems must resort to guess work, and are forced into over-reliance on human expertise to guide them. To ground the narrative in this paper, the examples of “robotics” and “identifying factually false information” are used to illustrate how many applications suffer from a lack of knowledge.

This raises important questions about AI infrastructure. What kind of knowledge resource is most needed in the modern context? What categories of knowledge should be included in such a resource? How should they be represented? How should they be made accessible to external users and applications? How can recent technological advances be harnessed to create such a knowledge resource? How should such a resource be evaluated?

A workshop at the 2025 Conference for the Advancement of Artificial Intelligence (AAAI) gathered over 50 researchers, including the authors of this paper, to explore these questions. This paper is an attempt to synthesize our discussions, and present a community view on the requirements and approach for creating a new knowledge infrastructure. We concluded that building an open engineering infrastructure for disseminating and using knowledge modules in practical applications, leveraging steady advances in knowledge representation and reasoning over the last 40 years, is a promising path forward. This includes establishing sets of conventions and social structures that should be adopted by contributors.

We begin by envisioning the type of knowledge resource required in the current context of AI, and then exploring aspects of creating it, including foundational knowledge, domain-specific knowledge, automated reasoning, and evaluation. We also review the current state of education in knowledge representation and reasoning. We conclude by presenting a community view on productive steps towards the creation of this much needed knowledge resource.

2 Envisioning a Knowledge Resource

We use the term *knowledge resource* to refer to a body of curated knowledge that can be examined and verified by humans. This knowledge could be formalized in any computational framework including ontologies [59], rules [28], a constraint network [20], a probabilistic causal model [60], and even in unambiguous natural language [48]. We envision the knowledge resource needed for AI by exploring four points: the type of problems that could best be solved by such a knowledge resource, the limits of current practice, what should be done differently, and the likely beneficiaries of such a resource.

2.1 Problems solvable by a Knowledge Resource

Data-driven learning is central to modern artificial intelligence (AI). But in some cases, curated knowledge can be better. For example, in basic arithmetic, people easily outcompete chatbots such as ChatGPT because they learn how to add and multiply numbers using rules rather than by looking at a large number of examples [14].

Curated knowledge works better than data-driven learning in these scenarios: rules needed to solve the problem are readily available, such as in systems of axioms for qualitative spatial reasoning [16, 24]; in applications that require high levels of accuracy, transparency, and reliability such as income tax calculations; and in contexts in which clarity and precision are crucial, such as in education.

A set of concrete examples where a knowledge resource can supplement the knowledge in large language models has been put out by Wolfram|Alpha [82]. These include “distance between Chicago and Tokyo”, “3 to the power 73”, “circumference of an ellipse with axes 4 and 12”, etc.

A knowledge resource is necessary for us to develop automated reasoning and intelligence that is more human-like, capable of participating in formulation of formal models from everyday inputs and interpreting their results in real-world terms (*model formulation* and *model interpretation*[24]). This requirement is supported by a recent survey of AI community in which 61.8% of survey participants estimated the minimal percentage of symbolic AI techniques required for reaching human-level reasoning to be at least 50% [66].

2.2 Limits of the current practice

Current knowledge resources such as Wikidata [78] and Google’s knowledge graph [69] capture relationships between real world entities such as people, places, organization, etc. ConceptNet captures relationships between concepts (for example, a knife is used for cutting) [70], but the relationships are limited to triples that are awkward, or insufficiently expressive, for capturing general-purpose knowledge. Large Language Models (LLMs) encompass enormous amounts of knowledge implicitly. This is effective for many applications but their output lacks formal guarantees [66].

Applications such as biomedicine require a much higher level of expressivity. For example, a drug may up-regulate or down-regulate a gene depending on the context [76]. The knowledge in biomedicine is often below the level of certainty of “established facts.” In clinical practice, a great deal of reasoning has to rely on assertions, statistical associations and observations. Existing reasoning engines are unable to accommodate varying levels of certainty at the discretion of the practitioner.

Knowledge resources such as Cyc [49], Wolfram|Alpha [38], Component Library [4] or its variant CoreALMLib [39] attempt to bridge this gap. Cyc, for example, targets formalization of complex knowledge patterns across the full spectrum of the human world. Because of the restricted intellectual property of Cyc and Wolfram|Alpha, limited access to them is not conducive to fostering an open-source community. OpenCyc, which is the open-source version of Cyc, is useful for limited purposes as it provides access to only a small fraction of the full knowledge base.

2.3 What should be different about a modern knowledge resource?

A modern knowledge resource should provide a public-utility-like infrastructure becoming a go to place for trusted and verified knowledge and reasoning methods across a variety of domains. Towards that end, it should provide knowledge in such a way that its formal representation is paired with its provenance in multiple modalities including text, images, video, or other formats such as graphs. As most pieces of knowledge are not universally true, emphasis must be placed on the applicable context of that knowledge.

The knowledge resource should advance the state-of-the-art in how knowledge representation and reasoning is leveraged in engineered systems. As there are ongoing initiatives assembling knowledge in data graphs (e.g., Wikidata uses RDF), and ontologies (eg., Stanford’s BioPortal uses OWL), we are envisioning that the proposed resource will use a language more expressive than these existing efforts. Examples of such languages include answer set programming [51] and its query-driven variants such as s(CASP) [2], Rulelog/Ergo [32] and other extended well-founded logic programs, theorem proving systems such as Lean [54], constraint representation [20], methods to incorporate uncertainty [46] and causal information [60, 61].

The knowledge resource should foster a distributed community, document use cases, and enable collaboration. Instead of wasting time debating ideal knowledge representations, the community’s focus should be on developing shared infrastructure and interoperability, and on maintaining consistency across representations, whenever possible. The resource should enable unanticipated linkages across diverse domains, for example, environmental knowledge could be cross linked to health knowledge; population knowledge could be linked to economic factors; etc.

2.4 Beneficiaries of a Knowledge Resource

The knowledge resource must exist in the modern ecosystem of societal needs and challenges.

The knowledge resource will benefit AI engineers creating novel AI applications by weaving together different off-the-shelf components. If knowledge modules are made available in a way that engineers can use them as easily as installing a new Python library, a range of AI applications will benefit including those incorporating agentic workflows and generative AI models.

The knowledge resource will also be an important new form of data for training machine learning systems. Most existing machine learning data sets are limited to facts.

The knowledge resource will benefit applications that require world models. Examples of such applications include those from the robotic planning community which must bridge between sensor-level data and higher-level cognitive world models, and the fact-checking systems that must detect factually false information.

With the growth in biomedical literature, it is not humanly possible for any one person to absorb and synthesize everything that is being published. A knowledge resource will enable superhuman knowledge aggregation and consistency maximization over complex domains (such as molecular biology, medicine or social policy) [81].

A knowledge resource will benefit knowledge representation and reasoning researchers by catalyzing research and application in this area. UC Irvine’s library of machine learning tasks and data sets [75], and the Hugging Face repository of transformer models [37] have catalyzed the research and adoption in machine learning and natural language processing. A similar open-source resource will benefit knowledge representation and reasoning researchers.

3 Formalizing foundational knowledge

Establishing foundational knowledge involves representing abstract knowledge such as knowledge about time, space, actions, causality, and mid-level knowledge such as the working of physical devices, or tenets of social psychology, qualitative physics, etc. [19]. To achieve maximum reusability and applicability, the knowledge resource must leverage foundational knowledge. We organize our discussion along four dimensions: easily available foundational knowledge, methodology for collecting and using foundational knowledge, evaluating its impact on the system behavior, and some long-term challenges.

3.1 Easily available foundational knowledge

As there has been much work in representing foundational knowledge, a knowledge resource can easily bootstrap from this prior work. This includes qualitative representations of time and space [16, 79, 24], representations of events and actions [27], formalizations of psychology [30], and representations to capture constraints [20] and probabilistic knowledge [60, 61, 46, 18]. OpenCyc’s ontology with integrated representations for vision, space, and language is also available [23].

3.2 Methodology for collecting foundational knowledge

The success of some past foundational knowledge representation efforts can be attributed to the rare contributions of ontological prodigies, for example, Pat Hayes, Jerry Hobbs and Ernest Davis. In contrast, a scalable approach requires community-curated resources that leverage repeatable processes and well-defined engineering best practices for building foundational knowledge infrastructure. To support this, a structured, multi-step approach is desirable.

First, start with a broad theoretical understanding by identifying key concepts in a domain. This process can be aided with corpus analysis, for example, by identifying sample texts in a domain for the occurrence of certain concepts [13].

Second, the dual goals of broad coverage and inferential competency in foundational theories can be pursued via the approach of successive formalization [30]. In this method, first-draft axioms are authored to support inference across the entire set of domain concepts previously identified in the first step, then incrementally formalized into competent logical theories through elaborations and refinements. The process of representation is iterative in that the initial choices for predicates and functions will likely evolve as the work progresses. It is crucial to recognize both obvious and subtle issues and to balance theoretical perspectives with practical problem-solving. Through this iterative process, one can develop a more refined understanding.

Finally, large language models can potentially contribute to the creation of foundational knowledge. This is, however, an area of research that is under-explored [36]. An initial discussion on the use of LLMs appears in Section 5.2.

3.3 Evaluating foundational knowledge

We can evaluate foundational knowledge using both intrinsic and extrinsic methods. Intrinsic evaluation involves a theoretical analysis based on criteria such as completeness, consistency, elaboration tolerance, redundancy, etc. Extrinsic evaluation involves showing that the use of foundational knowledge improves performance on a suite of tasks. Extrinsic evaluation is more easily understood by the stakeholders because of its direct connection to specific problems. Foundational knowledge can also be evaluated on the basis of whether it can inform the creation of nuanced and efficient test datasets. For example, instead of creating a broad test data set in which all examples are similar, we can use the foundational knowledge to identify the corner cases, and then design the test data to cover those corner cases, thus yielding a more compact test set. Finally, an indirect measure of the usefulness of foundational knowledge is in its adoption and reuse beyond its original creators.

3.4 Long-term challenges in formalizing foundational knowledge

There is no dearth of problems when it comes to formalizing foundational knowledge, but a few of them stand out.

First, real-world problems tend to be multi-modal in nature. For example, foundational knowledge will need to handle text combined with images, or mix audio, video and text. The foundational knowledge must address conceptual, temporal, and spatial aspects taken together.

Second, there needs to be a better bridge between the foundational knowledge and the abstract and vague aspects of natural language. Different logical forms of verbs, and different senses of words, can be typically understood by humans through surrounding context, but the current foundational theories offer no similar mechanisms. For example, consider the sentences "the bottle contains wine" and "the wine contains alcohol". Foundational theories must provide a way to disambiguate between such different uses of "contains".

Finally, there is still no easy way to translate across different representations of knowledge. For storing pictures, for example, there are multiple formats such as JPEG, PNG, etc. Translation tools exist for going across them, although some of the translations can be lossy. In contrast, knowledge represented in one particular formalism remains locked into that formalism, and it is not straightforward to exploit it in a system different from the one it was originally developed in. Creating a standard for interoperability between formalisms, such as was done for the less expressive language OWL, is a possible approach.

4 Automated Reasoning

The word "reasoning" has been used to refer to a variety of computational processes [66]. On one hand, we have deductive or probabilistic reasoning in which typically there is a formal proof that relates a question to its answer. On the other hand, we have inductive reasoning and analogical reasoning in which there may not always exist a formal proof that relates a question to an answer. In contrast to classical forms of deductive and inductive reasoning, a major emphasis in logic and knowledge representation and reasoning (KR) research has also been on abductive reasoning, which is a cornerstone of hypothesis formation and belief revision. Computer scientists strive to associate formal properties and guarantees for all of these reasoning processes. Examples of such formal properties include soundness, completeness and tractability.

For the purpose of the present paper, the term reasoning refers to the full spectrum of computational processes that have been assigned this label in the literature. Given this broad notion, we will organize our discussion along the following dimensions: discovering axioms automatically, modeling human reasoning, reasoning at scale, and incorporating context into reasoning.

4.1 Discovering Axioms

The problem of automatically discovering axioms has been traditionally studied under the topic of inductive logic programming [56]. Related efforts exist in qualitative reasoning to learn new patterns based on analogical generalization [52] or as default rules [80]. Modern practice, however, relies on a combination of manual and automatic approaches. For example, while pursuing Bayesian causal reasoning, it is expected that partial knowledge about the world is provided by an external source. Likewise, inductively acquiring knowledge about everyday embodied human interactions (e.g., from multimodal data) requires support for specialized domains such as space, time, and motion [72]. To support reliable and predictable decision support, large language models are coupled with an external module that contains human verifiable and explicit knowledge.

4.2 Open challenges in modeling human reasoning

Human reasoning is often prone to error. For instance, people frequently favor conclusions that are psychologically appealing over those that are mathematically sound [74]. This raises concerns about designing AI systems that directly mimic human reasoning as they risk inheriting these flaws. A more pragmatic objective is to develop AI systems that assist and augment human reasoning, helping individuals arrive at more accurate and reliable conclusions. In the following section, we outline the key challenges in pursuing this goal.

First, it will be helpful to develop a taxonomy of distinct kinds of human reasoning. Such a taxonomy will enable better communication about which aspect of human reasoning is being modeled in a computational system. A partial taxonomy is available in the existing literature for reasoning tasks such as query answering, planning, projection, diagnostics, etc.

Second, the human understanding of the world has a symbolic structure, and AI programs must exploit this to reason correctly even though they might rely on raw data for some of the processing. For example, when humans engage in diagrammatic reasoning, they are able to supplement their purely logical reasoning with diagrams and sketches by exploiting the symbolic structure of the world.

Third, just as humans can resolve ambiguities and conflicts during a conversation, reasoning tools must be able to do the same. Humans often make judgment calls because of these ambiguities and conflicts. Reasoning tools should be able to represent such judgment calls.

Fourth, the reasoning systems must be good at ignoring irrelevant details. For example, in traditional procedural programming a local variable has effect only in a certain scope. Similarly, these reasoning processes should be able to keep only a subset of facts in scope that are relevant for the current problem.

Fifth, the reasoning tools should be such that complex reasoning mechanisms that humans use, such as deduction, abduction, induction, counterfactual reasoning, etc., can be elegantly captured.

Finally, graphical models, be them deterministic or probabilistic enable causal and counterfactual reasoning that are central to human reasoning. A primary challenge is to acquire the causal model, even partially (namely the causal graph). Once a partial causal model is available, computing causal effect and counterfactual reasoning will be facilitated and should be further explored.

4.3 Reasoning at scale

With the growing knowledge in science, it is humanly impossible for any one person to effectively reason with it all at once. Reasoning at the scale of all science is, therefore, a practically useful challenge for AI systems. Reasoning can be particularly effective in processing what is already known about science, and pinpointing gaps for further research. For example, structural causal models and constraint reasoning models can be especially effective to support drug discovery and protein design.

4.4 Incorporating context into reasoning

Incorporating real-world constraints into reasoning is necessary for it to work correctly. Cyc’s knowledge base achieved this goal by organizing its knowledge into microtheories into a hierarchical structure [49]. The same goal can be achieved through other methods.

We must not assume our knowledge to be a single monolithic structure. Different knowledge modules that apply to different contexts should be able to interoperate with each other depending on the problem at hand.

Real-world reasoning scenarios themselves present constraints that the reasoning process should be able to pick up. For example, designing an artifact requires understanding its operating conditions to ascertain what materials are appropriate. Many of these issues can be addressed by checking inconsistencies with the real-world constraints, and through the development of specialized but domain-independent solvers integrating aspects such space, motion, actions, events, and dynamics [79, 71].

As science is a social process, reasoning methods must gracefully integrate with existing workflows and take into account the assumptions that the scientists are making. For example, a recent experiment combining qualitative process theory with a language model led to substantial improvements in performance [77].

Finally, as much of reasoning in science adopts a certain point of view, choosing the correct viewpoint is crucial. As an example, the shape of the Earth is viewed differently in topography versus astronomy.

5 Knowledge Curation using manual methods, machine learning and Large Language Models

Knowledge curation is the process of gathering, extending, and maintaining knowledge. Knowledge curation encompasses the full life cycle of specifying the requirements, schema design, data cleaning and loading, debugging and troubleshooting, and revising the knowledge.

Current practice on knowledge curation relies on teams of knowledge engineers and domain experts. We believe that this needs to change. Next we address the role of human oversight and automation in knowledge curation, and popularizing knowledge curation among scientific communities.

5.1 Role of human oversight in knowledge curation

Human oversight is indispensable in any knowledge curation effort. We will illustrate this using three use cases: knowledge curation to support web search, compliance to emission standards, and development of machine learning solutions.

Accurate results for web search require durable semantics. For example, a movie can go through several stages — initial publication of the story as a novel, availability of screenplay, filming and production, box office release, Netflix release, release in other languages, etc. As this process unfolds over a number of years, the search engine must correctly correlate different versions of the movie. At present, such correlation requires human oversight through a careful schema design.

To test the compliance of emission standards of automobiles, much sensor data is available, but most of it is not relevant. Human oversight is needed to identify the relevant aspects of sensor information that should be used for checking compliance to standards.

Most machine learning approaches require data that must be annotated by humans. Once the model is trained, *reinforcement learning with human feedback* is a vital part of model fine-tuning. Once a machine learning model is deployed, human oversight is necessary to ensure that the model performance does not drift as the input data evolves.

In summary, human experts can provide initial domain knowledge artifacts and define their intent to guide automated knowledge curation systems. This human-guided initialization helps systems understand specific domain contexts and requirements before any automation begins. Even after the initial automated generation of logical structures, human reviewers must validate and refine generated representations, ensuring accuracy and alignment [1]. The quality of automatically curated knowledge ultimately depends on human expertise to verify that automated outputs correctly represent the intended domain knowledge.

5.2 Role of Large Language Models in knowledge curation

We assume that any step of the knowledge curation process that could be automated should be automated. The question we address is whether previously human labor-intensive tasks can now be automated with

the advent of LLMs? LLMs can enable knowledge curation in at least three ways: becoming a source of knowledge, aiding in knowledge elicitation, and serving as knowledge curators.

LLMs capture an immense amount of knowledge implicitly. We can interrogate them to explicitly emit their knowledge on topics of interest for a given application. Such explicit knowledge, either in a natural or a formal language, can be used in multiple ways. It can be directly built into the application under human oversight and used by a reasoning process. It can also be used as a way to gain insights into the domain of interest which can speed up the downstream design work of curators.

LLMs have enabled the construction of powerful chat bots. This capability could be leveraged towards creating a systematic methodology to facilitate interdisciplinary knowledge acquisition. LLM-based natural language dialogs would need to be designed that support a domain expert in articulating knowledge which can be further curated by knowledge engineers or by the LLM itself.

LLMs might be used as components in knowledge curation interfaces between humans and large complex knowledge resources. As knowledge resources get big, they become difficult to understand by casual users. LLMs could provide an ability for a broader class of users to add and contribute their knowledge to a knowledge resource. In this scenario, an AI system using an LLM is a useful mediator between a human and a complex knowledge base.

In practice, frontier knowledge curation systems powered by LLMs have shown promise in analyzing documents, identifying key concepts, translating natural language into formal representations, and combining them into comprehensive knowledge models [1]. This automation significantly reduces the manual effort traditionally required for knowledge formalization. These systems can auto-formalize in order to validate claims, applying automated reasoning to detect factual inaccuracies with minimal human intervention and with explainability built-in. When validation fails, advanced systems can generate suggestions showing alternative representations that would resolve inconsistencies, effectively automating parts of the knowledge refinement process that previously required extensive human expertise.

We posit that an optimal approach to knowledge curation combines human expertise with automated systems in a continuous feedback loop. In this paradigm, humans provide initial knowledge artifacts and domain expertise, while automation handles complex transformations into formal structures that support verification [1]. Human-in-the-loop testing and validation exemplify this partnership: humans review and pose test scenarios and evaluate presented outcomes, while automated systems apply rigorous validation against established, formalized knowledge bases. When inconsistencies are detected, these systems provide grounded explanations and suggestions, which humans can then use to refine knowledge representations or improve system responses. This creates a continuous improvement cycle where human oversight guides automation, and automation enhances human capabilities.

5.3 Popularizing knowledge curation among scientific communities

The knowledge curation enterprise must be popularized among the scientific computing communities. Database curation efforts already exist across multiple sciences, by one count, there are 27 databases in materials science and chemistry alone [6]. But, few scientific communities are currently creating knowledge bases using expressive knowledge representation languages. Though some work leverages knowledge graphs [68, 55], current approaches have many limitations. For example, the representation of relationships between equations, variables, and broader theories that we see in Wikidata is limited. For instance, in Wikidata, “Stoke’s Theorem” is a “generalization of” “Green’s Theorem” – this captures some connections, but “generalization of” carries no deeper meaning about the nature of this generalization. Languages such as Lean provide greater expressiveness for formalizing scientific knowledge [7, 73], but its complexity makes it challenging to use by users outside formal logic disciplines.

Nonetheless, projects like PhysLean aim “to create a library of digitalized physics results in the theorem prover Lean 4, in a way which is useful to the broad physics community” [42]. We imagine formalized versions of scientific texts like Feynman’s Lectures [22], where the scientific content and derivations are structured, functional, and executable, with concepts interlinked across the text. Achieving this requires popularizing knowledge curation among scientific computing communities so that more of these specialists come forward to contribute.

6 Modern Education on Knowledge Representation

As highlighted in a recent report of a Dagstuhl seminar [21], there has been a consistent decline in the open academic positions in knowledge representation and reasoning, as well as in the number of students and researchers attracted to this field. There is a concern that after the current faculty members teaching knowledge representation retire, there is no plan to replace them. In this section, we consider in more detail the current practice for teaching knowledge representation, identify what is missing, and outline potential steps for the future.

6.1 Current practice for teaching Knowledge Representation

At most universities, especially in the United States, knowledge representation is taught as part of either an AI course or as a module on logic embedded in a course on discrete mathematics. Some universities provide knowledge representation and reasoning courses as advanced electives, and there are several textbooks to support such courses [9, 64, 35]. Currently used standard AI textbooks [67, 62] have several chapters on knowledge representation.

Several modern textbooks are available on logic programming [28, 51, 26, 25, 18]. Three of these textbooks focus exclusively on answer set programming [51, 26, 25]. They are used in courses taught by faculty members associated within that field. Two of these books come with online repositories containing slides and other teaching materials [43, 63]. There is also a textbook that focused on teaching an audience without technical background how to think using computational ideas [47] which has found a home in some philosophy courses. In addition, Computer Science departments at many universities offer courses in computational logic that cover knowledge representation and reasoning.

Wright State University has created an educational hub to teach people about knowledge graphs [44]. Their curriculum is tailored to different audiences ranging from students to senior executives. They are also developing an industry certification for knowledge graph professionals in cooperation with the Knowledge Graphs Conference.

Northwestern University has a Knowledge Representation and Reasoning course that exposes students to logic, Semantic Web technologies, and Cyc-style knowledge bases. Students get hands-on experience with industrial-scale knowledge bases with a heavy project-based component.

Within the industry, especially for the Cyc KB, teaching materials tailored to the technology have been developed. This teaching material typically assumes familiarity with logic. and focuses on teaching practical skills for expressing a given piece of knowledge formally.

6.2 Deficiencies in the current education practice

As recent AI research has been dominated by approaches based on machine learning, the coverage of knowledge representation in standard textbooks [67, 62] is out of date. There is also a tendency to portray the topic in less than positive terms.

The current teaching practice on knowledge representation does not always succeed in conveying its importance for building reliable computing systems. Most courses, with a few exceptions, are limited to using small theoretical examples, without making adequate connections to the real-world problems and actual impact. There is inadequate integration of logic into other computer science courses. For example, many students do not realize that they could use propositional logic to debug their if-then-else statements.

Another major missing piece in the current teaching practice on knowledge representation is in cultivating an ability to identify implicit and explicit knowledge and rules of thumb that capture how the world works. In other words, the current teaching fails to cultivate skills to conceptually model a task domain and answering questions about what that task/domain is. For example, given a short story such as: “For sale: baby shoes, never worn”, one should be able to infer the implicit possibility of the death of the child and the tragedy of having to sell the shoes. Such skills are cultivated in courses on philosophy and literature, but similar skills are needed for computer science students to be effective at knowledge engineering.

There is also a philosophical deficiency in the framing of computer science curricula as they primarily focus on “how to build” vs “how to understand”. This encourages students to rush towards coding solutions

instead of developing clear specifications. Clear specifications require them to think formally about the requirements and clarify any implicit information before engineering a solution.

6.3 Steps to improve the teaching of Knowledge Representation

Knowledge Representation and Reasoning community should undertake an initiative directed to make the teaching materials easily available. Much can be learned from a similar effort undertaken at the University of California at Berkeley for a course on introduction to AI [5].

To make the teaching materials easy to use, the community should develop teaching modules that can be easily picked up and plugged into a variety of computing courses. The teaching modules should touch on various aspects of knowledge representation, and be accompanied by slides, worked out examples, sample exercises, sample projects, and exam questions.

Consistent with these goals, the Prolog Education Group (PEG) was founded in 2022, on the occasion of the 50th anniversary of the programming language Prolog, to “teach logic, programming, sound reasoning, and AI” to people of all ages and to develop and provide relevant educational resources [17]. The efforts of the PEG group need to be expanded beyond teaching programming to include topics of knowledge representation.

Modern platforms for instructors such as Gradescope [31] or GitHub Classroom [29] support means to implement automatic grading utilities for programming exercises and projects. Providing such utilities is indispensable for easy adaptation of teaching materials.

Similarly, providing user-friendly sandbox environments online for various modules will lower the barrier for integrating material at different academic levels. For example, while undergraduate students in a computer science program might be asked to install a specific reasoner or solver to execute sample code, an online sandbox environment supported by that reasoner could allow a high school student to practice the same task.

New materials need to be developed to address “why care” questions that illustrate the practical application of knowledge representation in multiple fields. A repository of real-world examples should be developed.

We should integrate teaching of logic programming into other computing courses. For example, within a course on databases, logic programming should be taught as the foundation of modern database management systems. Similarly, a course on programming languages can incorporate material depicting answer set programming and the algorithmic aspects of systems that support this knowledge representation paradigm. Some universities, for example, University of Nebraska, Omaha, and University of Texas at Dallas, already integrate logic programming into other courses as suggested here.

The community should develop a *knowledge representation body of knowledge* similar to *software engineering body of knowledge* [8], against which certifications could be granted. Teaching materials should also underscore the importance of interoperability, demonstrating how different knowledge representations and reasoning systems can be integrated. A key component of the teaching materials should be clear instructions and examples of how to integrate the knowledge resources into different applications.

Much of R&D in knowledge representation over the last 2–3 decades has focused on extending the expressiveness of declarative logic programs to go beyond that of knowledge graphs and relational databases. AI education should include coverage of those key expressive features, including answer set semantics versus well-founded semantics, higher-order syntax, constraints, etc.

AI literacy courses are being developed across many institutions to teach people about new developments. The community should engage with such initiatives to ensure that the role and value of knowledge representation is adequately covered. It is especially important to include an adequate history of different developments to avoid reinvention of already-known concepts.

It would be valuable to introduce set theory and logic in high schools [12]. These students should be taught different forms of reasoning including deduction, abduction and induction. The importance of logic should be highlighted at an early age through games, puzzles and debates. This will not only make students critical thinkers, but also alleviate any fear of logic courses in college. The Prolog Education Group has initiated efforts in this direction [34].

Most courses in computer science, except perhaps software engineering, can be re-framed as knowledge representation courses by focusing on objects, their relationships, and the types of questions or tasks associated with them. Even in software engineering, precise reasoning is critical for developing requirement

specifications. The key distinction between an introductory programming course and a knowledge representation course lies in the type of knowledge being represented, its intended audience, and its impact. By embedding knowledge representation throughout the curriculum, students could be trained as both computer scientists and implicit knowledge representation experts—without explicitly realizing it. This approach reframes computer science not just as an engineering discipline but also as a natural science.

More broadly, universities can be encouraged to develop an explicit categorization of courses relevant to knowledge representation (perhaps in computer science, philosophy, library science, etc.). Universities could be encouraged to offer a minor concentration in knowledge representation. A consolidated resource that gathers different courses and their relevance to knowledge representation would make it easier for universities to offer this minor.

7 Evaluating a Knowledge Resource

We can think of evaluating a knowledge resource in at least three different ways. First, we can evaluate individual modules, for example, evaluating foundational knowledge, or evaluating how effectively existing knowledge sets are hosted and disseminated. Second, we can evaluate it in terms of how effectively it fosters a community that helps create it. Finally, we can evaluate a knowledge resource as an enabler of AI and its role in the current frontier of AI developments.

This section situates the knowledge resource evaluation in the context of current developments of AI. From that perspective, we will first consider the limitations of the current practice of evaluations in AI, and then consider a few alternative ways to perform better evaluations through expert interviews, virtual environments, and examination of the working of the system.

7.1 Limitations of the current evaluation practice

Current evaluation practice in AI suffers from proxy failure—when a measure becomes a target it ceases to be a good measure [41]. With human testing there is an imperfect correlation between a benchmark and the underlying capability that the benchmark is assumed to measure; with AI evaluation this gap is much wider. For example, a program doing well on a multi-state bar exam is hardly suitable to practice law. Furthermore, when benchmarks are the basis for evaluating and comparing AI systems, those benchmarks are subject to corruption pressures, such as training to the test.

The Winograd schema challenge [45] was designed to evaluate the ability to use commonsense knowledge to disambiguate pronoun references. The research community has developed programs that do well on the test without incorporating the explicit common sense that the test was originally meant to test.

We next discuss different approaches to address proxy failure.

7.2 Evaluation through expert interviews

Evaluation of an AI system by an expert user interacting with it over just a few hours can be much more insightful than the quantitative metrics reported by benchmarks [15]. Unlike the famed Turing test, the evaluator must know that the subject of evaluation is a computer program, and should be given information about its architecture. If the task requires domain expertise, the evaluation panel can include both an AI expert and a human expert.

7.3 Evaluation in virtual environments

Virtual environments, simulated worlds and games can be effective in evaluating targeted aspects of reasoning. For example, the angry birds competition [65] has been effective at exploring reasoning about actions and qualitative reasoning. Game environments and cognitive robotics tasks can also constrain the allowed moves in ways that can explicitly force any successful program to reason.

7.4 Evaluation of the functioning of the system

Benchmarks should test not just the output of the system but the reasoning steps and individual pieces of knowledge that were used in producing the result. Even though it may be expensive to produce such test sets from scratch, existing resources such as the Cyc knowledge base could be leveraged to generate such tests. Such a resource would significantly enhance the existing collection of problems assembled by the common sense reasoning community

8 Next steps towards creating a knowledge resource

There was an overwhelming consensus at the workshop that a meticulously curated knowledge resource along with the necessary tools and methodologies for its effective use is sorely needed.

For creating such a knowledge resource, much can be learned from the Hugging Face repository [37]. A similar dynamic platform promoting interoperability among various knowledge representations and reasoning systems should be created. The portal should be organized around specific tasks that can be performed using the knowledge. It should provide a sandbox for trying out different reasoning capabilities without requiring any licenses or difficult installations. The portal should inspire contributions from students taking knowledge representation courses world-wide. The problems addressed by the portal should be of immediate relevance to a cross section of the industry. It should be straightforward to make use of any given package by simply doing "pip install <package-name>".

While some argued for intrinsic value of a knowledge resources on its own. Standalone knowledge resources already exist, for example, Common Logic Ontology Repository [33], Standard Upper Merged Ontology [58] and Basic Formal Ontologies [3]. These prior existing knowledge resources differ from the Hugging Face model, and the new knowledge resource envisioned here, in that they do not target any specific end-user task. In addition to focusing on specific tasks, the success of Hugging Face can be attributed to providing interoperability between a few prevalent deep learning models, creating a unified API that simplifies training across different model architectures and tasks, and providing a hub for model training and discovery.

The knowledge resource should be positioned as a source of trusted and verifiable knowledge. LLMs should be explored as an initial use case needing a trusted knowledge resource. In addition to LLMs, a few other use cases must be identified. These use cases must span the breadth of knowledge levels/certainty, as well as mission criticality.

To avoid some of the barriers that have prevented Cyc from being widely incorporated in contemporary AI systems across academia and industry, there was an overwhelming consensus that the knowledge resource needs to be open-source and released under a permissive license.

The community also agreed on the need for effective teaching materials in the form of modules that can be easily adapted by others. More details have been discussed in Section 6.3.

Acknowledging the importance of evaluation and validation, our discussion highlighted the need for test sets and benchmarks, potentially developed in collaboration with existing initiatives such as Cyc and MLCommons [53]. This would ensure the reliability and accuracy of the knowledge resources, while also mitigating the risk of redundant or inaccurate data proliferation. The creation of public test sets, useful for evaluating the performance of different systems, would be a valuable teaching tool.

A strong consensus emerged that the resource should be created and/or managed by a nonprofit foundation, allowing for membership from both academic and commercial entities. This structure would ensure sustainability and broad participation. To kickstart the initiative, a virtual institute was suggested as an initial phase.

9 Conclusion

Much work needs to be done to turn the community vision articulated here into an actionable research program. In addition to the initial steps suggested in the previous paragraph, the importance of research in all aspects of creating the knowledge resource is equally important. Significant research is needed in methodologies for creating both foundational and domain-specific knowledge, in effective reasoning techniques that

scale and take context into account, and in translational techniques that can automatically translate a reasoning task framed in one formalism into another formalism. Enabling seamless distributed development, especially by domain experts is essential for fostering an effective community.

AI research has always moved in cycles, with certain paradigms rising and falling in prominence. The remarkable progress we now see in deep learning would not have been possible without the persistence of researchers like Geoffrey Hinton, who championed neural networks long before they were in vogue. Just as neural networks have proven powerful for modeling aspects of human perception and cognition, curated knowledge remains essential for capturing structured, declarative understanding — the kind that underpins reasoning, learning, and communication. Although knowledge curation has receded from the forefront of mainstream AI, its value has not diminished. By investing sustained effort into re-integrating curated knowledge into the AI toolkit, we can unlock new forms of robustness, interpretability, and societal impact — and, once again, broaden the horizons of what AI can achieve.

10 Acknowledgments

This work has been supported by grant number 2514820 from the United States National Science Foundation. We are grateful to Prof. Ernest Davis for leading the panel discussion on foundational knowledge at the TIKA workshop.

References

- [1] Adewale Akinfaderin and Nafi Diallo. Minimize generative ai hallucinations with amazon bedrock automated reasoning checks. AWS Machine Learning Blog, April 2025. Posted 1 April 2025.
- [2] Joaquin Arias, Manuel Carro, Elmer Salazar, Kyle Marple, and Gopal Gupta. Constraint answer set programming without grounding. *Theory and Practice of Logic Programming*, 18(3-4):337–354, 2018.
- [3] Robert Arp, Barry Smith, and Andrew D Spear. *Building ontologies with basic formal ontology*. Mit Press, 2015.
- [4] Ken Barker, Bruce Porter, and Peter Clark. A library of generic concepts for composing knowledge bases. In *Proceedings of the 1st International Conference on Knowledge Capture, K-CAP '01*, page 14–21, New York, NY, USA, 2001. Association for Computing Machinery.
- [5] UC Berkeley. Uc berkeley cs188 intro to ai – course materials. <https://ai.berkeley.edu/home.html>, 2025. Accessed April 9, 2025.
- [6] Blaiszik. awesome-matchem-datasets: A curated list of datasets for materials chemistry. <https://github.com/blaiszik/awesome-matchem-datasets>, 2025. Accessed April 9, 2025.
- [7] Maxwell P. Bobbin, Samiha Sharlin, Parivash Feyzishendi, An Hong Dang, Catherine M. Wraback, and Tyler R. Josephson. Formalizing chemical physics using the Lean theorem prover. *Digital Discovery*, 3(2):264–280, 2024.
- [8] Pierre Bourque and Richard E. Fairley, editors. *Guide to the Software Engineering Body of Knowledge (SWEBOK), Version 3.0*. IEEE Computer Society, Los Alamitos, CA, 2014. Available online.
- [9] Ronald Brachman and Hector Levesque. *Knowledge representation and reasoning*. Elsevier, San Francisco, CA, 2004.
- [10] Ronald J Brachman and Hector J Levesque. Toward a new science of common sense. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12245–12249, 2022.
- [11] Vinay Chaudhri, Chaitanya Baru, Naren Chittar, Xin Dong, Michael Genesereth, James Hendler, Aditya Kalyanpur, Douglas Lenat, Juan Sequeda, Denny Vrandečić, et al. Knowledge graphs: introduction, history and, perspectives. *AI Magazine*, 43(1):17–29, 2022.

- [12] Vinay K Chaudhri. The u.s. should teach kids to think logically, 2024. Accessed: 2025-04-03.
- [13] Vinay K Chaudhri, Nikhil Dinesh, and D Incezan. Creating a knowledge base to enable explanation, reasoning, and dialog: Three lessons. *Advances in Cognitive Systems*, 3:183–200, 2014.
- [14] Vincent Cheng and Zhang Yu. Analyzing ChatGPT’s mathematical deficiencies: Insights and contributions. In Jheng-Long Wu and Ming-Hsiang Su, editors, *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 188–193, Taipei City, Taiwan, October 2023. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).
- [15] Anthony G Cohn and Jose Hernandez-Orallo. Dialectical language model evaluation: An initial appraisal of the commonsense spatial reasoning abilities of llms, 2023.
- [16] Anthony G Cohn and Jochen Renz. Qualitative spatial representation and reasoning. *Foundations of Artificial Intelligence*, 3:551–596, 2008.
- [17] Veronica Dahl. Prolog Education Group, Accessed: 4/21/2025.
- [18] Adnan Darwiche. *Modeling and reasoning with Bayesian networks*. Cambridge university press, Cambridge, 2009.
- [19] Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015.
- [20] Rina Dechter. *Constraint processing*. Morgan Kaufmann Publishers, San Francisco, CA, 2003.
- [21] James P Delgrande, Birte Glimm, Thomas Meyer, Mirosław Truszczyński, and Frank Wolter. Current and future challenges in knowledge representation and reasoning, 2023.
- [22] Richard Feynman, Robert B. Leighton, and Matthew Sands. The feynman lectures on physics. <https://www.feynmanlectures.caltech.edu/>, 2013. Accessed: April 9, 2025.
- [23] Ken Forbus. NextKB: An Integrated, Linguistically, Visually, and Inferentially Rich Open-license Knowledge Base, Accessed: 4/16/2025.
- [24] Kenneth D Forbus. *Qualitative representations: How people reason and learn about the continuous world*. MIT Press, Cambridge, 2019.
- [25] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. *Answer Set Solving in Practice*. Morgan & Claypool Publishers, Switzerland, 2012.
- [26] Michael Gelfond and Yuliya Kahal. *Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach*. Cambridge University Press, Cambridge, UK, 2014.
- [27] Michael Gelfond and Vladimir Lifschitz. *Action languages*. Linköping University Electronic Press, Linköping, 1998.
- [28] Michael Genesereth and Vinay K Chaudhri. *Introduction to logic programming*. Morgan & Claypool Publishers, San Francisco, 2022.
- [29] GitHub. Github classroom. <https://classroom.github.com/>, 2025. Accessed April 10, 2025.
- [30] Andrew S Gordon and Jerry R Hobbs. *A formal theory of commonsense psychology: How people think people think*. Cambridge University Press, Cambridge, UK, 2017.
- [31] Gradescope. Deliver and grade your assessments anywhere. <https://www.gradescope.com/>, 2025. Accessed April 9, 2025.

- [32] Benjamin Grosz, Michael Kifer, Theresa Swift, Paul Fodor, and Janine Bloomfield. Ergo: A quest for declarativity in logic programming. In David S. Warren, Veronica Dahl, Thomas Eiter, Manuel V. Hermenegildo, Robert Kowalski, and Francesca Rossi, editors, *Prolog: The Next 50 Years*, pages 224–236, Switzerland, 2023. Springer Nature (LNAI 13900).
- [33] Michael Grüninger, Torsten Hahmann, Ali Hashemi, Darren Ong, and Atalay Özgövde. Modular first-order ontologies via repositories. *Applied Ontology*, 7(2):169–209, 2012.
- [34] Gopal Gupta, Elmer Salazar, and Joaquín Arias. Computational thinking with logic programming. In *Proc. 2nd ICLP’24 Prolog Education Group Workshop*, volume 3799 of *CEUR Proceedings*, Dallas, TX, 2024. CEUR-WS.org.
- [35] James Hendler, Fabien Gandon, and Dean Allemang. *Semantic web for the working ontologist: Effective modeling for linked data, RDFS, and OWL*. Morgan & Claypool, Waldham, MA, 2020.
- [36] Pascal Hitzler, Andrea Nuzzolese, Catia Pesquita, and Cogan Shimizu. Large language models for knowledge graph and ontology engineering. <https://kastle-lab.github.io/llms-and-kg-engineering/>, 2024. Accessed: April 24, 2025.
- [37] Hugging Face. Hugging face, 2025. Accessed: April 3, 2025.
- [38] Wolfram Research, Inc. Wolfram—one, Version 14.2, 2025. Champaign, IL, 2024.
- [39] Daniela Inclezan. Corealmib: An alm library translated from the component library. *Theory and Practice of Logic Programming*, 16(5–6):800–816, 2016.
- [40] Yuqian Jiang, Fangkai Yang, Shiqi Zhang, and Peter Stone. Task-motion planning with reinforcement learning for adaptable mobile service robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019)*, November 2019.
- [41] Yohan J John, Leigh Caldwell, Dakota E McCoy, and Oliver Braganza. Dead rats, dopamine, performance metrics, and peacock tails: proxy failure is an inherent risk in goal-oriented systems. *Behavioral and Brain Sciences*, 47:e67, 2024.
- [42] Joseph Tooby-Smith. PhysLean: Digitalizing Physics in Lean 4, 2025. Accessed: April 9, 2025.
- [43] Yuliya Kahl and Michael Gelfond. Additional resources for the 2014 textbook by gelfond and kahl. <http://pages.suddenlink.net/ykahl/>, 2025. Accessed April 10, 2025.
- [44] Kastle Lab. Education gateway, 2025. Accessed: 2025-04-03.
- [45] Vid Kocijan, Ernest Davis, Thomas Lukasiewicz, Gary Marcus, and Leora Morgenstern. The defeat of the winograd schema challenge. *Artificial Intelligence*, 325:103971, 2023.
- [46] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, Cambridge, MA, 2009.
- [47] Robert Kowalski. *Computational logic and human thinking: how to be artificially intelligent*. Cambridge University Press, Cambridge, UK, 2011.
- [48] Robert Kowalski, Jacinto Davila, Galileo Sartor, and Miguel Calejo. Logical english for law and education. In David S. Warren, Veronica Dahl, Thomas Eiter, Manuel Hermenegildo, Robert Kowalski, and Francesca Rossi, editors, *Prolog - The Next 50 Years*, number 13900 in LNCS, pages 287–299, Switzerland, 2023. Springer.
- [49] Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.
- [50] Douglas B. Lenat, Mayank Prakash, and Mary Shepherd. Cyc: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine*, 6(4):65–85, 1985.

- [51] Vladimir Lifschitz. *Answer set programming*, volume 3. Springer Nature Switzerland, Switzerland, 2019.
- [52] Matthew D McLure, Scott E Friedman, and Kenneth D Forbus. Learning concepts from sketches via analogical generalization and near-misses. In *Proceedings of the annual meeting of the cognitive science society*, volume 32, pages 1771–1776, Portland, OR, 2010. Cognitive Science Society.
- [53] MLCommons. Mlcommons, 2025. Accessed: 2025-04-03.
- [54] Leonardo de Moura and Sebastian Ullrich. The lean 4 theorem prover and programming language. In *28th International Conference on Automated Deduction*, pages 625–635, Switzerland, 2021. Springer.
- [55] David Mrdjenovich, Matthew K Horton, Joseph H Montoya, Christian M Legaspi, Shyam Swaraknath, Vahe Tshitoyan, Anubhav Jain, and Kristin A Persson. propnet: A Knowledge Graph for Materials Science. *Matter*, 2(2):464–480, February 2020.
- [56] Stephen Muggleton. Inductive logic programming. *New generation computing*, 8:295–318, 1991.
- [57] National Academies of Sciences, Engineering, and Medicine. *Nobel Prize Summit: Truth, Trust, and Hope: Proceedings of a Summit*. National Academy of Sciences, Washington, DC, 2023.
- [58] Ian Niles and Adam Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9. ACM, 2001.
- [59] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.
- [60] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, San Francisco, 1989.
- [61] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, 2nd edition, 2009.
- [62] David L Poole and Alan K Mackworth. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, Cambridge, 2010.
- [63] Potassco. Course on answer set solving in practice. <https://teaching.potassco.org/>, 2025. Accessed April 10, 2025.
- [64] Raymond Reiter. *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT press, Cambridge, 2001.
- [65] Jochen Renz, Xiaoyu Ge, Stephen Gould, and Peng Zhang. The angry birds ai competition. *AI Magazine*, 36(2):85–87, 2015.
- [66] Francesca Rossi, editor. *AAAI 2025 Presidential Panel on the Future of AI Research*. Association for the Advancement of Artificial Intelligence, Washington, DC, 2025. Accessed: 2025-04-16.
- [67] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Pearson, Harlow, UK, 2021.
- [68] Marwin H. S. Segler and Mark P. Waller. Modelling Chemical Reasoning to Predict and Invent Reactions. *Chemistry – A European Journal*, 23(25):6118–6128, May 2017.
- [69] Amit Singhal. Introducing the knowledge graph: things, not strings, May 2012.
- [70] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, pages 4444–4451, San Francisco, CA, 2017. American Association for the Advancement of Artificial Intelligence.

- [71] Jakob Suchan and Mehul Bhatt. Semantic question-answering with video and eye-tracking data: AI foundations for human visual perception driven cognitive film studies. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2633–2639, New York, 2016. IJCAI/AAAI Press.
- [72] Jakob Suchan, Mehul Bhatt, and Carl Schultz. Deeply semantic inductive spatio-temporal learning. In James Cussens and Alessandra Russo, editors, *Proceedings of the 26th International Conference on Inductive Logic Programming (Short papers), London, UK, 2016*, volume 1865 of *CEUR Workshop Proceedings*, pages 73–80, London, UK, 2016. CEUR-WS.org.
- [73] Joseph Tooby-Smith. HepLean: Digitalising high energy physics. *Computer Physics Communications*, 308:109457, March 2025.
- [74] Amos Tversky and Daniel Kahneman. Judgment under uncertainty: Heuristics and biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *science*, 185(4157):1124–1131, 1974.
- [75] UCI Machine Learning Repository. UCI Machine Learning Repository, 2025. Accessed: April 3, 2025.
- [76] Deepak R. Unni, Sierra A. T. Moxon, Michael Bada, Matthew Brush, Richard Bruskiewich, John H. Caufield, Paul A. Clemons, Vlado Dancik, Michel Dumontier, Karamarie Fecho, Gustavo Glusman, Jennifer J. Hadlock, Nomi L. Harris, Arpita Joshi, Tim Putman, Guangrong Qin, Stephen A. Ramsey, Kent A. Shefchek, Harold Solbrig, Karthik Soman, Anne E. Thessen, Melissa A. Haendel, Chris Bizon, Christopher J. Mungall, and The Biomedical Data Translator Consortium. Biolink model: A universal schema for knowledge graphs in clinical, biomedical, and translational science. *Clinical and Translational Science*, 15(6):e13302, June 2022.
- [77] Barres Victor, McFate Clifton, Kalyanpur Aditya, Saravanakumar Kailash Karthik, Moon Lori, Seifu Nati, and Bautista-Castillo Abraham. From generating answers to building explanations: Integrating multi-round rag and causal modeling for scientific qa. In *Proceedings of NAACL 2025 - Industry Track*, Albuquerque, 2025. NAACL.
- [78] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.
- [79] Przemyslaw Andrzej Walega, Mehul Bhatt, and Carl Schultz. ASPMT(QS): non-monotonic spatial reasoning with answer set programming modulo theories. In Francesco Calimeri, Giovambattista Ianni, and Mirosław Truszczyński, editors, *Logic Programming and Nonmonotonic Reasoning - 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings*, volume 9345 of *Lecture Notes in Computer Science*, pages 488–501, Lexington, KY, 2015. Springer.
- [80] Huaduo Wang and Gopal Gupta. FOLD-SE: an efficient rule-based machine learning algorithm with scalable explainability. In *Proc. PADL*, volume 14512 of *LNCS*, pages 37–53, Switzerland, 2024. Springer.
- [81] Michael Witbrock, Karen Pittman, Jessica Moszkowicz, Andrew Beck, Dave Schneider, and Douglas B Lenat. Cyc and the big c: Reading that produces and uses hypotheses about complex molecular biology mechanisms. In *AAAI Workshop: Scholarly Big Data*, pages 27–32, Austin, TX, 2015. AAAI.
- [82] Stephen Wolfram. Wolfram—alpha as the way to bring computational knowledge superpowers to chatgpt, January 2023. Stephen Wolfram Writings.