International Journal of Approximate Reasoning 136 (2021) 308–324

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Some thoughts on knowledge-enhanced machine learning

Fabio Gagliardi Cozman *, Hugo Neri Munhoz

Escola Politécnica — Universidade de São Paulo, Brazil

ARTICLE INFO

Article history:
Received 21 June 2020
Received in revised form 18 April 2021
Accepted 7 June 2021
Available online 24 June 2021

Keywords:
Knowledge representation
Machine learning

ABSTRACT

How can we employ theoretical insights and practical tools from knowledge representation and reasoning to enhance machine learning, and when is it worthwhile to do so? This paper is based on an invited talk delivered at ECSQARU2019 around this question. It emphasizes the knowledge representation and reasoning side of knowledge-enhanced machine learning, looking at a few case studies: the finite model theory of probabilistic languages, the generation of explanations for embeddings, and an "explainable" version of the Winograd Challenge.

1. Introduction

This paper is based on an invited talk delivered at the 2019 edition of the *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*. We thank the organizers of the conference for their invitation to have this material in print. The goal of that talk was to offer a few selected ideas on "knowledge-enhanced machine learning". The topic had a rich history before 2019, and since then it has moved to center stage within Artificial Intelligence (AI) research. Even though the astonishing speed of AI research is turning ideas that were original by the end of 2019 into clichés at the beginning of 2021, we hope to have something useful to say by focusing on a few handpicked examples.

Indeed, there has been increasing interest in the combination of classic AI programs with data-driven machine learning. Take for instance the invited talks at the *2020 AAAI Conference on Artificial Intelligence*,[1] where many debates mentioned the interaction between reactive and deliberative behavior, respectively associated with the System 1 and System 2 modes of thought popularized by Kahneman [52]. Often these modes of thought, the first faster and automatic, the second slower and effortful, were compared to AI systems without and with explicit reasoning engines. A highlight of that conference was the invited talk by Henry Kautz, with an analysis of AI's current summer and special attention to neuro-symbolic combinations. As noted by Kautz, there is "violent agreement on the need to bring together the neural and symbolic traditions." But there is no consensus on how to do it. For instance, one might enhance neural networks with the ability to take rules as inputs or to produce rules as outputs. But it would be perhaps too narrow to focus solely on neural networks; these are surprisingly powerful but still there are many other animals in the zoo of machine learning techniques. Actually, the broad challenge upon AI is to combine high level representation and reasoning with data-driven learning. One can probably conceive in excess of a million ways to mix such things.

In this paper we do not attempt to review, or even to classify, all possible combinations of knowledge-based and data-driven approaches. For instance, we do not discuss the many neural architectures that emulate symbolic reasoning — we offer only a short overview of those efforts. Likewise, we do not do justice to the myriad existing techniques that discuss

* Corresponding author.
E-mail address: fgcozman@usp.br (F.G. Cozman).

[1] Several talks are now at https://aaai.org/Conferences/AAAI-20/livestreamed-talks/.

https://doi.org/10.1016/j.ijar.2021.06.003
0888-613X/

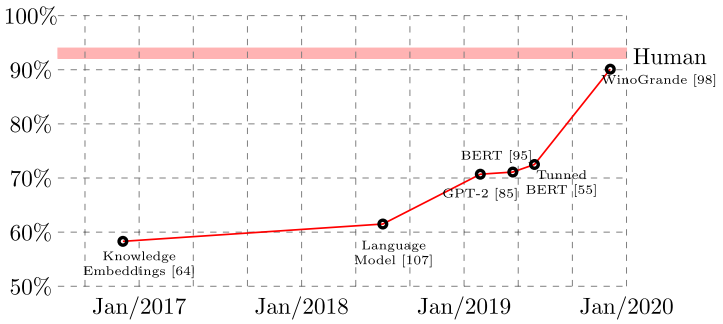


Fig. 1. Accuracy of recent systems on the Winograd Challenge [55,64,85,95,98,107].

how to build models in established formalisms, say logic programming, by learning them from data. Rather, we examine a few strategies where knowledge representation and reasoning can enhance machine learning, thus adding to recent literature calling for more integrated AI research. Of course the boundaries of our theme are quite subjective, as it is difficult to draw the limits both of knowledge representation and reasoning and of machine learning.

We start, in Section 2, with a very brief comment on the current success of AI and in particular machine learning. We discuss key questions in Section 3: What is knowledge representation and reasoning good for, in the presence of so much data? What are the avenues that have been explored in mixing it with machine learning? How can we use insights from knowledge representation and reasoning to enhance machine learning?

We then focus on three specific settings, examining ways in which knowledge representation and reasoning can be valuable. First, in Section 4 we briefly look at the importance of expressivity and complexity analysis in the context of probabilistic languages. In Section 5 we look at a few issues in the construction of explanations and their reliance on background knowledge. And in Section 6 we discuss the Winograd Challenge, its goals, and a possible extension that includes explanations. While we do not report on novel results in Sections 4 and 5, in Section 6 we present a new approach to Winograd Scheme explanation that employs linguistic representation and reasoning. By going through such case studies, we isolate patterns where data-driven machine learning can be enhanced by knowledge-based reasoning. We share a few concluding thoughts in Section 7.

2. Hot summer in AI

There is now unprecedented optimism concerning what AI can do; previous AI winters are long gone [80]. In particular, data-driven machine learning has seen extraordinary success. Perhaps the best summary of the situation is captured by the expression “the unreasonable effectiveness of data” coined by Pereira, Halevy, and Norvig in 2009 (already more than ten years ago!). Their paper [45] discusses the fact that a large bulk of raw data can be used to produce remarkably intelligent behavior. One example they mention is machine translation, an activity that defied linguistics in previous decades,[2] and that is now quite accurately done by fitting large nonlinear models to huge corpora.

Here is another example of the unreasonable effectiveness of data, an example to which we will return later in more detail. The Winograd Challenge consists of a number of Winograd Schemes [62], where one is asked to solve pronoun disambiguation questions such as: [116]:

The city councilmen refused the demonstrators a permit because they advocated violence. Who are “they”?

The Winograd Challenge has been proposed as a replacement for the venerable Turing Test [63]: to verify whether a machine is intelligent, we should check whether it can tackle the Winograd Challenge as well as a human subject. No dull pattern-reproducing machine should get close to the more than 90% accuracy displayed by human subjects in solving the ambiguities found in Winograd Schemes.[3] And indeed for several years, between 2012 and 2017, machines would hardly go much beyond 50% accuracy. However, that suddenly changed when powerful language models became reality. Fig. 1 shows the astonishing recent evolution in accuracy. Machines now display accuracy higher than 90% in the Winograd Challenge, in essence by learning huge inscrutable language models from giant datasets! This outcome embodies the current supremacy of data-driven machine learning within AI. One may be led to believe that machines must tirelessly learn until they somehow capture all possible knowledge.

[2] AI researchers who worked in the eighties and nineties probably remember the joke where the statement “the spirit is willing, but the flesh is weak” is automatically translated to Russian and back to English, just to produce “the vodka is good, but the meat is rotten”; even if apocryphal, the joke seemed quite credible back then (it even appears in a respected AI textbook [90] and in the New York Times edition of April 28 1983 within a text titled *The Computer as Translator*).

[3] The reported average human performance in the Winograd Challenge ranges from 92.1% [6] to 94% [98].

3. Why knowledge representation and reasoning?

Given recent developments in data-driven machine learning, one might ask, is knowledge useful at all? As stated, this question seems too dramatic, and a bit naive. In AI, the word "knowledge" has a broad meaning often including uncertain and nonmonotonic statements.[4] With such a lenient understanding, knowledge is (trivially) useful; even a neural network mapping audio in English to text in French captures beliefs about the relation between inputs and outputs. Any artificial intelligence carries knowledge in its own particular form, and "AI-style knowledge" is perhaps uncontroversially useful.

The more pressing question then is whether knowledge representation and reasoning (KRR) is useful at all. To be precise: Do we need structural ingredients that affect behavior and that can be taken as declarative accounts of the stored knowledge,[5] together with explicit reasoning mechanisms that operate on them? Note that we do *not* contemplate here a return to old-style knowledge-based expert systems; we merely ask whether it is profitable to resort to tools of KRR in practical circumstances.

There are some answers in the literature as to why KRR is useful. A common point is that a well-designed KRR-based system should be easy to debug, modify, and explain [11, Section 1.2]; it should also be a system that can tackle open-ended challenges. The latter point may be too optimistic; sometimes a representation puts a straightjacket over a modeling session, limiting the possible paths to follow. However, it is certainly correct to say that knowledge allows one to explore infinite paths that are potentially encoded in it, thus allowing more abstract reasoning.

Today perhaps the strongest argument for KRR is an economical one. If something is already known, it is better to archive it in suitable form rather than to leave it to be learned anew. We can benefit from background knowledge when starting up our systems so that they can learn more quickly and from less data. One can use background knowledge to bias data processing techniques in useful ways (to resort to some jargon, background knowledge can be used as inductive bias [4,72]). This is particularly important if the background knowledge consists of rules that prevent discriminatory or harmful behavior; such rules may be subtle and may be based on ethical or legal arguments that are naturally declarative. Moreover, once we learn something, we can certainly transfer and reuse it; it should be easier, and cheaper, to pass around a few thousand beliefs than a multi-billion parameter model. Thus it makes sense to learn declarative elements that can be partially reused.

Also, it takes computing power to process all the data and parameters; so much so that there is now concern about energy consumption in machine learning [105]. Not only computing power is needed; considerable human expertise, and quite a bit of patience, is needed to test various hyperparameters and configurations, and to run experiments that may take hours or days. Previous knowledge can be useful in guiding the process. Of course we can learn every aspect of a difficult problem by collecting data, and in so doing we may even get our algorithms to reach super-human ability; however, it takes effort to do such data collection. Besides, in many practical circumstances one is faced with small datasets as it may be too costly to obtain additional observations. Few-shot learning is often needed due to monetary constraints, but in some cases it may even be impossible to get as much data as one wants due to ethical or legal constraints.

In another direction, KRR can simplify the connection between AI devices and their human designers and users. We can imagine a science fiction world where all machines behave in ways that they learned solely from data. These machines might interact successfully amongst themselves without any need to resort to symbols. But if humans are also present in that world, then surely machines will have to agree to communicate with them using fully formed concepts and relationships. In particular, consider a machine that must explain its decisions. Perhaps these decisions are produced by an input-output relationship that was learned from data. Of course the machine can also learn to generate explanations, and then learn how to explain how it learned to explain, and so on. However, presumably the explanation process will be smoother if the original decision is based on some reasoning that can be then expounded (we return to explanations later). Again, data-driven machine learning may reach super-human heights but it demands considerable resources to do so.

Moreover, even if machines can somehow learn extremely complex mappings from inputs to outputs, mappings that are so detailed that they produce excellent decisions, an exclusive focus on end-to-end learned solutions does not seem to be wise from an engineering point of view. Typically large systems consist of parts that can be isolated and understood, fixed or replaced if needed; modularity is a key (human?) design strategy. The designer of such a system must get to know how the various parts of the system behave and interact through high level concepts.

In any case, given the current state of matters in AI, as sketched in Section 2, it is wise to assume that KRR will help by *enhancing* rather than by replacing data-driven learning. To summarize, here is a list of possible reasons as to why KRR can enhance machine learning: KRR can help in (1) inserting existing expertise into a problem as built-in resources; (2) imposing a necessary inductive bias to start learning; (3) coding rules that prevent discriminatory/unethical behavior; (4) debugging and modifying an artificial intelligence; (5) explaining and justifying decisions, as well as communicating concepts, to human users; (6) organizing large systems in ways that make sense to human designers; (7) transferring facts

[4] Technical papers in AI are quite lax about the word "knowledge", and even handbooks and textbooks adopt a very generous perspective on "knowledge", sometimes taking care to write "commonsense knowledge" or "human knowledge" [11,24,87,113], and moving away from the philosophical focus on justified true belief [2]. We might perhaps replace "knowledge" by "belief" and then say "belief representation", "background beliefs", etc; alas, "belief" itself is already a heavily overloaded term in AI.

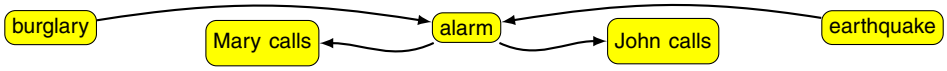
[5] This sentence is inspired by Smith's *knowledge representation hypothesis* [87]; he used the expression "propositional account" instead of "declarative account", but in the context of AI one should focus on accounts that are declarative (that is, consciously representable).

and rules learned in a context to another one; (8) teaching a machine through formalized instructions; (9) learning with less data and less energy consumption. On top of these, KRR can help in: (10) providing a solid grasp of foundational questions that require formal analysis, a point we emphasize in Section 4.

The next three sections discuss some of the points made in the last paragraph, with a focus on the question: How to enhance data-driven machine learning with knowledge representation and learning? We cannot deal with every aspect of this question, so we focus on three case studies where KRR enhances machine learning techniques. As already noted, in Section 4 we look at the contribution of KRR in providing tools for formal analysis. Then in Section 5 we examine settings where explanations for recommendations based on embeddings are generated using symbolic manipulation. And in Section 6 we further discuss knowledge-based explanations in the context of question answering.

Before we get into these more specific sections, it pays to survey some of the most important connections already made between knowledge representation and machine learning. There are many possible perspectives; for instance, with multi-relational learning one detects patterns in relational databases, using relational versions of say decision trees [30,31]. Much more relevant to our discussion is the work on Inductive Logic Programming (ILP), where the main goal is to extract, from a given set of facts, a logic program that entails the set of positive facts and does not entail the set of negative facts [74,75]. Clearly, ILP satisfies some of the desiderata raised in the previous paragraphs, as ILP can benefit from previous knowledge, can lead to transferable results, can generate explanations. Indeed, recent work has looked at "ultra-strong" learning, where the goal is not just predictive performance, but rather to deliver symbolic insights that teach a human user [76]. During the last thirty years the scope of ILP has been enlarged to encompass various logical forms and uncertainty measures; in particular, Probabilistic Inductive Logic Programming focuses on techniques that build, from data, logic programs where rules or facts may be associated with probabilities [27].

Another important connection between representation and learning has emerged in the literature on graph-theoretical probabilistic models. Bayesian networks, originated in KRR as "graphical representations of probabilistic knowledge" [81, title of Chapter 3], have become common targets of machine learning methods [58,77]. However, the representational power of Bayesian networks is limited by the fact that they are propositional in nature. A simple example illustrates this fact. Take the well-known Mary/John example [97]: an alarm may ring with probabilities that depend on a burglary or an earthquake taking place; if the alarm rings, there is a probability that Mary will make a call, and likewise for John. This example can be represented by a small Bayesian network with the following graph:



Many practical problems contain repetitive patterns that call for logical variables and the like (for instance, the popular Latent Dirichlet Allocation model [9]). Thus a variety of template-based representations languages appeared during the nineties to allow specification of large repetitive Bayesian networks [39,59,65,115]. A turning point, around 2000, was the introduction of learning techniques aimed at probabilistic relational models [35]. A large spike of interest in probabilistic relational models ensued (already a case where KRR tools where mixed to great effect with machine learning solutions). The many proposed models have been examined in books [26,27,38] and overview papers [16,54]. Some of the proposed languages are based on graphs, such as Probabilistic Relational Models [57] and Multi-Entity Bayesian Networks [19] while others rely on textual descriptions, such as Relational Bayesian Networks [49] and Bayesian Logic Programs [28]; some proposals are based on logic programs [34], while others pursue completely general programming languages [40]. The somewhat different but notably influential language of Markov Logic Networks [29,91] also combines explicit logical forms with probabilistic weights that are usually learned from data. In fact, the initial desire to expand the representational power of Bayesian networks morphed, at the end of the nineties, into a strong interest in learning models with logical and probabilistic elements. One usually refers to all of those proposals as belonging to Statistical Relational Learning, a label that clearly reflects sympathy towards machine learning.

Probabilistic ILP and Statistical Relational Learning, broadly understood, try both to enhance symbolic reasoning with data-driven methods, and to enhance machine learning with high-level machinery. We could select many ways in which KRR can make a difference in enhancing machine learning within this space; in Section 4 we focus on one issue that, perhaps due to its subjectivity, is rarely mentioned: the fact that the KRR culture of formal investigation can jazz up our understanding of the more sophisticated forms of learning.

A different combination of symbols and data is pursued under the label of Neuro-Symbolic Computing. This is not a new idea at all; authors often note that even McCulloch and Pitts landmark efforts on neural modeling touched on the connection between logical inference and neural networks [68]. As artificial neural networks received enormous attention from mid-eighties to nineties, several forms of neuro-symbolic integration surged as well. Since the explosive growth of deep learning after 2010 or so, Neuro-Symbolic Computing has also gone through frenzied times. Fortunately, there are now many excellent books, surveys and opinion pieces about Neuro-Symbolic Computing, some going back to ten to twenty years ago [3,12,21,22,46,47], others trying to capture the present moment [20,23].

There is no unique way to classify neuro-symbolic systems. One might divide them into Unification and Hybrid systems [21]: the former ones rely on connectionist components that perform symbolic computation, while the second ones have distinct connectionist and symbolic components. Both categories can be further divided. Unified systems may encode

each concept of interest through an artificial neuron (localist approaches) or through sets of artificial neurons (distributed approaches). Hybrid systems may be split into a variety of approaches, some where the various components only communicate results amongst themselves, some where the symbolic elements are translated into artificial neurons [47]. However, there are popular strategies that do not fit well into this classification scheme; for instance, we might define a category consisting of those connectionist systems that, during the training process, take into account symbolic input through their loss functions or through changes in their optimization paths. A different classification has been proposed by Bader and Hitzler, as they suggest a scheme that takes into account not only the unification/hybrid dimension but also the purpose of the neuro-symbolic system (representation or extraction of knowledge) and the formal language employed by the symbolic elements [3].

A more recent classification scheme has been sketched by Kautz in the AAAI2020 talk we alluded to in Section 1; we summarize here the adapted version explored by Lamb et al. [61] and expanded by Garcez and Lamb [20]. Kautz' first category is just a kind of deep learning: neuro-symbolic systems that start with symbols, embed them into a deep network, and match outputs with symbols. Some deep network architectures are particularly suited to such Type 1 systems; for instance, Graph Neural Networks can directly handle relations amongst objects [100]. Type 2 systems instead contain a symbolic problem solver that calls, as a subroutine, a connectionist component. The paradigmatic example is the AlphaGo system where tree search is guided by artificial neural networks. We can of course include here hybrid systems with many interacting symbolic modules that may call connectionist modules. Kautz next type is one where not only facts but also rules and logical formulas are translated into artificial neural elements that help in training a connectionist component. Many old and new techniques fall within this translational scheme; most of them translate propositional expressions but some resort to more complex logics (such as temporal or modal ones) or even mathematical expressions. Also we must include here recent attempts to cast the translation using fuzzy logic and similar languages [92,103]. Kautz then defines a type in which a neural front end processes input and sends its output to a symbolic reasoner, a pipeline that is popular today with applications that start from raw data (images, words) and must go through some high level reasoning or planning. And Kautz finishes with a type of neuro-symbolic systems where a connectionist component calls, as a subroutine, symbolic problem solvers. The latter systems are referred to as Type 6 by Lamb et al. [61]; they introduce a Type 5 category consisting of those systems where rules are mapped into embeddings, perhaps regularizing the loss function used to train a deep network.

Whatever the classification one chooses, one finds that most neuro-symbolic systems operate at a rather low level. In fact it is almost a digital level: neurons are built to emulate operations of propositional logic, memory elements, even Turing machines or simple programming languages. Apart from all the contemporary action around simple symbolic blocks handled by deep learning, there has been a persistent interest within Neuro-Symbolic Computing to enhance KRR by translating symbolic notions into embeddings and neural networks. The resulting models then run computations, sometimes by learning how to do inference, sometimes by converging to desired solutions.

Likewise, there has also been a steady interest within Neuro-Symbolic Computing to enhance neural networks with KRR tools, a perspective that is aligned with our focus in this paper. Since the inception of Neuro-Symbolic Computing there has been an effort to use background knowledge to simplify the construction of neural networks or to reduce the need for training data. More recently there has been renewed emphasis on the other end; that is, on the extraction of knowledge from neural networks. In particular, rule extraction can produce results that are interpreted by the human user, a theme that has received serious attention in Neuro-Symbolic Computing, as the most up-to-date overviews of the area reveal [20,23]. In short, there is much in KRR that can be used to enhance connectionist models, not only in supporting better and cheaper training, but also in explaining decisions and in aiding designers with formal analysis. We return to the latter point at the end of Section 4; in Sections 5 and 6 we return to matters of explanation and deal with settings where connectionist modeling has been quite successful.

4. Blending the KRR culture into machine learning

The "KRR culture" shared by researchers interested in KRR is, roughly speaking, one that values solid mathematical and philosophical formal investigation. The machine learning culture also resorts to heavy mathematics but it typically deals with statistical methods, dynamic systems, numerical optimization; moreover, it is sharply focused on performance measures such as accuracy. We do not plan to criticize either culture; rather, we wish to point out that the KRR way of thinking can enhance our understanding about the outputs of learning procedures.

We develop our point in the following subsections by looking at probabilistic models over relational interpretations. The presentation contains some shameless self-citation, but this is due to the fact that we want to focus on a topic we are familiar with.

4.1. The KRR perspective on probabilistic modeling

Section 3 briefly summarized the (large!) literature on languages that extend Bayesian networks with relational and logical elements. Some are based on graphs or frames, others on logic or functional programming [16]. This diversity is not a problem from a pragmatic machine learning perspective: it is reasonable to address different problems through different solutions, perhaps empirically testing various alternatives. However, some difficulties remain. For instance, what is exactly

the formal language of probabilistic relational models that one is trying to learn? What should such a model be able to express? What is it not able to express? What would be the cost of each intended computation?

Possibly Jaeger was the first to advocate that probabilistic modeling languages should be studied via KRR tools [50]. He proposed to import model-theoretic strategies to better navigate through the forest of modeling probabilistic languages. In particular, he examined which features of one language can be translated to another language, thus building a hierarchy of languages.

An established strategy in KRR research is to look at an abstracted small set of notions that captures a large space of possible languages. We can apply this strategy in the context of probabilistic modeling languages used in machine learning, so that we can ask general questions about complexity and expressivity. As much as KRR once helped by providing the formalisms behind probabilistic relational models, the KRR culture can suggest ways to enhance the analysis of these models.

To illustrate, in the next subsection we look at a possible way to capture a large set of probabilistic modeling languages and to study their complexity and expressivity [17].

4.2. Some finite model theory for relational Bayesian network specifications

Suppose we have a finite vocabulary of relations; for instance, we may have a vocabulary with relations burglarized and connects, where burglarized(x) indicates that x has been burglarized, and connects(x, y) indicates that x and y are connected.

Suppose also that each relation r may be associated either with an assessment or with a formula, as follows. An assessment is written $\mathbb{P}(r) = \alpha$, indicating that each grounding of r gets probability α; for instance, $\mathbb{P}(\mathsf{calls}) = \alpha$ means

$$\mathbb{P}(\mathsf{calls}(\mathsf{Mary})) = \alpha, \quad \mathbb{P}(\mathsf{calls}(\mathsf{John})) = \alpha,$$

and so on. The other possibility is to associate r with a formula using the syntax

$$r(x_1, \ldots, x_k) \equiv \phi(x_1, \ldots, x_k),$$

where ϕ is well-formed logical formula with free logical variables $x_1, \ldots, x_k$ with k the arity of r. We assume that no relation depends on itself if we follow a sequence of such statements (that is, the transitive closure of "depends" with respect to formulas is acyclic).

This abstracted language may seem restrictive but it is actually quite general; as a very simple example, consider defining a Gilbert-style random graph with a binary relation edge (where each instance of a logical variable is a node in the random graph) and determining whether it is fully connected [18]:

$$\mathbb{P}(\mathsf{edge}) = 0.1,$$

$$\mathsf{fully} \equiv \forall x, y : \neg(x = y) \rightarrow (\mathsf{edge}(x, y) \vee \mathsf{edge}(y, z)).$$

We refer to any consistent set of assessments and formulas following the syntax in the previous paragraphs as a *relational Bayesian network specification*. Note that to produce an actual Bayesian network out of such a specification, the set of groundings must be produced, so we must fix a set of individuals (for instance, {Mary, John}). The computation of the probability of some grounding, conditional on some other groundings, is an *inference*.

So we have an abstract formal language. What can we do with it? We can now ask what is the complexity of inferences as we vary the class of allowed formulas ϕ. To simplify the discussion, denote by $\mathcal{L}$ the language consisting of strings representing the allowed formulas.

We have that the computation of an inference is a PSPACE-complete problem, provided $\mathcal{L}$ is function-free first-order logic with a bound on relation arity [17, Theorem 10].[6] However, there are some languages, for instance derived from the DL-Lite description logic [14], that admit polynomial inference [17, Theorem 17]! And more: if we fix the specification, then the computation of an inference is an easier problem: for instance, inference is PP-complete when $\mathcal{L}$ is again function-free first-order logic with a bound on arity [17, Theorem 7]. This is important because it tells us that, if the specification is fixed, then inference costs as much as ordinary Bayesian network inference [94], and therefore similar algorithms may be of use. Still more: suppose we fix not only the specification but also the groundings used in the inference, and we only let the set of individuals be given as input; what happens to the cost of computing inferences? Nicely, we find that this question has equivalent versions in the theory of probabilistic databases [112] — an example of how an abstracted formalism helps one find bridges amongst existing topics. And we then find that if $\mathcal{L}$ is function-free first-order logic where at most two logical variables are used, then the calculation of the probability for some fixed expressions and grounding is actually polynomial, a truly remarkable result given that such a language is quite powerful [5].

Another issue that is dear to the KRR culture is expressivity. One possibility, already mentioned, is Jaeger's suggestion that we should compare the abstract features of languages so as to place them in a hierarchy [50]. But there is a different route, where we ask for an absolute measure of expressivity. This is the route followed by the theory of descriptive complexity [41],

[6] This result, as well as others mentioned in this paragraph, can be properly formalized by defining clearly what is meant by the various decision problems [17].

where one determines a class of Turing machines that can be perfectly emulated with a language of interest. When those machines exactly define a complexity class C, then we take the language to capture C. A key result in descriptive complexity is Fagin's theorem: the language of existential function-free second-order formulas captures NP [32]. This intriguing result says that the class NP, typically defined through polynomial-time nondeterministic Turing machines, can be defined in an entirely different way, solely by resorting to (function-free second-order) logical formulas.

Analogously, relational Bayesian network specifications capture the complexity class PP when ϕ can be any function-free first-order formula [18, Theorem 2].[7] The intuition behind this result is this. Suppose one has a social or physical phenomenon that gets some input and produces an output with some probability. This phenomenon can be simulated with a polynomial-time probabilistic Turing machine *if and only if* it can be represented by a relational Bayesian network specification restricted to function-free first-order logic.

More complex phenomena may require more powerful modeling languages. For instance, if one lets a single formula ϕ to be in existential function-free second-order logic, while the other formulas are still in function-free first-order logic, the resulting specification language does capture the complexity class $\mathsf{PP}^{\mathsf{NP}}$ [18, Theorem 3]. Even more freedom for formulas ϕ lets one capture even larger classes [18, Theorem 4].

Additionally, it is possible to employ techniques from finite model theory [42] to demonstrate inexpressibility results — that is, to prove that particular probabilistic models cannot be produced when ϕ is restricted to particular languages. To avoid a technical discussion we do not dwell on such matters here [18, Theorem 7].

4.3. The challenge, and the opportunity ahead

To summarize, the KRR culture can contribute not only with specific tools but with an overall approach that has been assembled through decades (in some cases, millennia…) of investigation. As we have indicated in this section, KRR techniques can be profitably applied to languages that are the targets of machine learning.

The challenge, and the opportunity, is then to bring this KRR culture to high dimensional curve fitting schemes, say by looking at the descriptive complexity of large random forests, or of artificial neural networks and the like. The latter seems to be a path filled with open questions. On the one hand, there are already results on the power of artificial neural networks to approximate functions and to do Turing complete computations [83], and results on the power of artificial networks to embed a number of temporal and modal logics [22] and logic programs [3]. On the other hand, there are many more formal languages that can be produced by combining logical features and that could, at least in principle, be captured (or not) by various networks and machine learning models: Which models can/cannot capture these languages, and at what computational cost? As an example, consider the recent elegant results on the expressivity of Graph Neural Networks [117]. These latter networks offer a balance between relational structure (the underlying graph) and learning (the functions associated with the nodes of the graph), so they live somewhere between the raw curve fitters and the symbolic representation schemes. A detailed study of their power, in line with the KRR culture, will certainly be a major accomplishment.

5. Explaining through dialogue

Assembling and communicating explanations is a key human ability, and it is not surprising that AI has paid attention to it for a considerable time. Back in 1981 a study with the influential expert system MYCIN found that users (physicians) considered a "system's ability to explain its advice to be its most important attribute" [104, Section 34.2.4]. In fact, a key argument in favor of expert systems was that their decisions could be explicitly explained [13, Section 2.4].

Explanations are not only useful to explain decisions; they are essential in human learning, where a student typically receives explanations from a teacher. And explanations are intimately related to knowledge: to quote from authorities, we find in Plato's dialogue *Meno* that "true beliefs are not worth much until you tie them down with a reasoned explanation."[8] Since Plato's writings, explanations have drawn attention from many fields, from philosophy to psychology and cognitive science. Fortunately we do not need to review all of this literature as an excellent survey on explanation theory from an AI perspective has been recently produced by Miller [70]. We can thus start from recent concern around explanations for machine learning methods.

5.1. XAI and KRR

Renewed interest in explanation facilities has sprung from fear that several successful machine learning techniques are too opaque: users may not be comfortable nor trust such algorithms; errors may not be understood; decisions may follow unethical/discriminatory patterns. There has been a call for more transparent and interpretable artificial intelligences, and one way to produce such devices is to have them explain their own behavior. The popular term EXplainable AI, often abbreviated XAI, has been coined by an influential DARPA research program [43].

[7] Again, this result, as well as others mentioned in this paragraph, can be properly formalized [18].

[8] We resort to Chappell's translation [15, Section 34c(ii)].

Understandably, XAI has energized the KRR base; for instance, many conferences connected to KRR have included XAI within their focus, and many have held related workshops. Current XAI focuses on ways to enhance machine learning by making it more interpretable, certainly a worthy goal.

However, not every technique in the surging XAI literature is related to KRR; in fact, perhaps most of them are not. Many current XAI techniques focus on determining which parts of a model [88] or an input [102] most influence the output, without any need for KRR.[9]

When is it the case that KRR can enhance current XAI? One opportunity can be found in conversational task-oriented agents that must get some useful work done (as opposed to just chatting). An agent may have to explain its own decisions; in doing so, the agent must take into account common knowledge and clear contrastive arguments [70]. Besides, successful explanations may require several exchanges with associated abductive reasoning and planning [48].

5.2. Explanation in conversational recommendation agents

To be concrete, consider conversational recommendation agents. We can expect users to have more trust in the agent if it provides not only recommendations but also explains them whenever requested [106].

Take a conversational recommendation agent whose recommendations are based on previous content and on similarity: if an item is known to be preferred, then the agent recommends other similar items [89]. For instance, suppose we have a conversational agent for students that can recommend classes: if a student asks for some topic, the agent will suggest classes that touch on that topic and related ones.

Now suppose the recommendation agent must actually explain the suggestion if so requested.

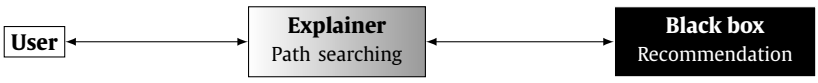
Many content-based recommendation systems learn similarities that are not easy to interpret. Why is it that a class on Forensic Engineering was suggested to a student interested in History? A correct explanation may be: because an embedding, learned from data by gradient descent, takes both concepts to close points in high dimensional space... but which student wants such an explanation?

This is a situation where background knowledge can be useful. For instance, both ExpLOD [78] and ASEMF-UIB [1] recommendation systems resort to knowledge graphs to produce explanations. They do so by looking for paths in a knowledge graph that connect the similar entities, building an explanation out of these paths. Note that a knowledge graph is just a store of triples of the form $< h, p, t >$ where the head h and the tail t are entities and p is a binary predicate [10,71]. A path from entity h to entity t is a set of triples $< h, p_1, e_1 >< e_1, p_2, e_2 > \cdots < e_{n-1}, p_n, t >$. As an example, ASEMF-UIB explains the fact that a user likes an actor by noting that the user likes movies where the actor appears [1, Figure 7]. Such an explanation moves away from trying to describe the inner workings of the machine learning system; it rather tries to capture the reasons behind the recommendation. Alas, the difficult with such knowledge-based methods is that even large knowledge graphs are markedly incomplete — that is, many links are not known to hold [79]. ASEM-UIB addresses this concern by applying inference rules [1], but such inference procedures take precious time from the conversation.

One might then take not the raw knowledge graph so as to generate explanations, but rather resort to links that are suggested by a state-of-art embedding applied to the original knowledge graph. That is, one may use graph completion techniques that employ embeddings learned from the original knowledge graph, where each entity is mapped to a point and each relation is mapped to a vector or similar object [118]. Typically such embeddings are generated through gradient-based optimization, often resorting to neurally-inspired architectures borrowed from deep learning. The embedding actually does all the work regarding the recommendation, but the result must be processed into a symbolic explanation.

This idea has been implemented in a conversational recommendation system that suggests classes [84]. As an example, the agent suggests a class on *Legal Engineering* to a student interested in *History*, explaining that *Legal Engineering* is a topic of *Law*, and both *Law* and *History* fall within the Humanities. Increases in trust and engagement have been observed in tests with the implemented conversational recommendation agent [84].

In a sense, the strategy just outlined is a "series" combination: we have a data-driven agent that is enhanced with extra knowledge/inference:



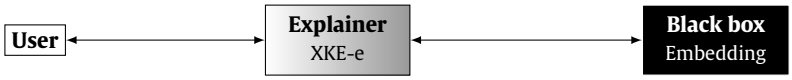
Now suppose that, instead of simply using an embedding to support knowledge-based explanations, one wishes to explain the embedding itself. Note that embeddings that explicitly encode relationships between entities may be rather opaque, so it is fair to ask: Why is it that the embedding determines two entities to be linked? This is another situation where we can enhance a data-driven technique by mixing it with symbolic operations.[10] We now sketch a strategy that combines embeddings with the Subgraph Feature Extraction algorithm, a path-building method [36] connected with logical inference [37].

[9] A very rich presentation of such XAI techniques is kept by Molnar [73].

[10] Alternatively, we might learn a neural network that produces explanations; maybe we can even imagine an infinite hierarchy of explainers. Finding data to learn such explainers will not be an easy matter; we again return to this point later when we discuss the Explanation Winograd Game.

To be concrete, we focus on the recent XKE-e algorithm [44,96], even though many variants are possible. We have as input a knowledge graph and an embedding learned from it; our goal is to explain why the embedding decides triples to be true or false. Note that we do not want to justify why a particular triple must be true (it may not be); we simply want to explain what is it that the embedding is doing, so that the user can trust it (or not). Note also that we are looking for a global explanation that captures the whole behavior of the embedding, and not a local explanation geared towards a particular triple.

The XKE-e algorithm uses Subgraph Feature Extraction to extract a large set of paths amongst entities from the knowledge graph, so as to detect the statistically frequent paths. Then the XKE-e builds a logistic regression that mimics, to the extent that it is possible, the decisions made by the embedding on the pairs of entities. Finally, XKE-e uses the coefficients of the logistic regression to determine which paths lead to a particular decision, thus explaining the decision in a particular sense. In short, XKE-e builds an approximation to the embedding by taking symbolic paths as features. Again, we have a "series" combination:



6. Solving the Explaining Winograd Challenge

As noted in the Introduction, the Winograd Challenge was conceived to advertise commonsense reasoning, but it has become one of the most dramatic victories of the machine-learning-can-do-everything philosophy. In this section we look at the Winograd Challenge as a laboratory for knowledge-enhanced machine learning, and we extend it by asking: How can one *explain* the solution of a given Winograd Scheme? We show that a bit of KRR can go a long way at a modest computational cost.

In Section 6.1 we build the rationale for an *Explaining Winograd Challenge*, and in Sections 6.2 and 6.3 we describe respectively the assumptions behind and the implementation of a minimal system that solves and explains a class of Winograd Schemes by exploring commonsense perception of causality.

6.1. The Explaining Winograd Challenge

The victory of pattern extraction techniques in attaining human performance in the Winograd Challenge is impressive because the challenge was designed to avoid statistical tricks. Each Winograd Scheme mentions two parties and contains a pronoun or possessive adjective that refers to one of the parties; there must be a SPECIAL word that determines the referent of the pronoun or the possessive adjective in such a way that, when the special word is replaced by an ALTERNATE word, the referent changes — presumably the pair of SPECIAL/ALTERNATE words makes it difficult to rely on "clever tricks" based on statistical correlation between words [63].

Yet today's most successful approaches are simple-minded: in essence they try both referents, checking which one is ranked higher by some learned large scale language model [56].[11] There is now a feeling that, with enough data, any Winograd Scheme will be cracked by pattern-reproducing machines.

One might think that better designed Winograd Schemes would eventually unmask the extend to which symbolic reasoning is required by intelligence. It does seem that the full power of Winograd Schemes has not been realized, as one can build, with some moderate effort, and at the cost of writing a long sentence, a Winograd scheme whose solution does require serious logical thinking. Consider an example[12]:

> Donna took 4 red stamps and 4 green stamps, threw two of them away, and affixed two of them in the back of each of three logicians Ann, Bella, and Carol, so that each logician could only see the stamps in the other two logician's backs; Miss Marple, who could not see any of the stamps, asked Ann, Bella and Carol in this sequence whether they knew the colors of their own stamps, and they replied No, No, No in sequence — and when Miss Marple asked Ann again whether she knew the colors of her own stamps, and she replied YES/NO, she concluded that one logician surely had distinct stamps in her back, and she smiled at her. Who is the last "her" referring to?

A Winograd Challenge made of riddle-like sentences may indeed defeat the most subtle pattern extractor. But then, such a Challenge may defeat human subjects, too; in the end, by moving to very difficult Winograd Schemes we may make things easier for the computer than for the human, leaving miserable humans befuddled and humiliated.

[11] The WinoGrande dataset, containing 44,000 schemes, was inspired by the original Winograd Schemes and constructed by crowd-sourcing. A language model (RoBERTa) trained in the WinoGrande dataset reached 90.1% accuracy in the Winograd Challenge.

[12] This scheme was inspired by a "Riddle of the Week" in *Popular Mechanics* [7].

It seems more appropriate to look at variants of the original Winograd scheme that exercise more dimensions of intelligent behavior, moving away from a narrow focus on accuracy without going all the way to the Turing Test.

Here is a proposal: spice up the Winograd Challenge by demanding explanations to be provided. We can envision a human judge that chooses a Winograd Scheme and gets in return not only a solution to the scheme but an explanation for the solution (the human judge would then rate the explanation as acceptable or not). This is much more constrained than the Turing Test but much more challenging than the Winograd Challenge. A similar Turing-like Test has been advocated by Schank [101]: in his *Explanation Game*, both a machine and a person get a specific task, and the human judge asks both about their solutions; the judge determines which answers are more insightful and explanatory, and if the machine beats the human, then the machine is said to understand at some prescribed level (determined by the task). It does make sense to have a graded test of intelligence; similarly, the Explaining Winograd Challenge is relative to the difficulty of the proposed Winograd Schemes and the expected ability of the human judge.

We submit that explanations based on linguistic structure offer a promising strategy to address the Explaining Winograd Challenge. Consider a Winograd Scheme:

The lawyer asked the witness a question, but he was reluctant to ANSWER/REPEAT it. Who is "he"?

Clearly, if we choose ANSWER, then the pronoun "he" refers to the witness, but why? Obviously, the lawyer is not answering the question: one party asked a question and now the expected social convention is that the other party is to do something. Here is an abstract version of this explanation: if A acts (asking) upon B, then B is expect to react to A (answering). Now if we choose REPEAT, the abstract explanation is: A acts (asking) upon B and then A does something else not really directed at B (repeating). That is, the explanation is embedded in the linguistic structure imposed by verbs and other connectives; what matters is the structure of the social conventions as captured by language. Of course, transforming this explanation frame into a human-friendly text is a secondary matter once the frame is found, so we do not worry about the final formatting step.

To elaborate, consider another Winograd Scheme that seems harder to explain:

Pete envies Martin BECAUSE/ALTHOUGH he is very successful. Who is "he"?

If we choose BECAUSE, the abstract explanation is: if A holds a feeling about B, the cause of this feeling is expected to say something about B. If instead we choose ALTHOUGH, the abstract explanation is: if A holds a feeling about B, and the cause of this feeling is not expected to say something about B, then it must say something about A. This dry explanation frame may not be rhetorically alluring, but it can be easily turned into more convincing text through simple templates. The point is that explanation frames explain *why is it that the pronoun was disambiguated the way it was.*

This explanation strategy is not the only possible one. For instance, the Winograd Scheme solver by Trinh and Le [107] finds the word most likely to produce a particular solution; one can concoct an explanation that just indicates which word led to the decision. Alternatively, one might try to rephrase an input Winograd Scheme counterfactually, trying to cajole the reader into accepting the solution. For instance, the explanation may state that "the city councilmen refused the demonstrators a permit because the demonstrators advocated violence; had the permit been approved, the demonstrators would cause harm". To borrow terminology from Psychology, these latter explanations are *horizontal* [51,53]; we do not think they are as effective as the causally-based explanation frames we have discussed in previous paragraphs.

We are not aware of any attempt to address the Explaining Winograd Challenge; in the next two subsections we present a rather simple solver that resorts to existing knowledge bases for linguistic reasoning. We first analyze in more detail the Winograd Challenge (Section 6.2) and then we present the design and test of our solver for a class of Winograd Schemes (Section 6.3).

6.2. Classifying Winograd Schemes

When we look at existing Winograd Schemes, we see that they are usually built within one of two distinct contexts. First, we may have a folk psychology context ruled by knowledge of social conventions (as in schemes discussed in the previous subsection). Second, we have have a naive physics context combined with factual knowledge.

Consider yet another example of *social* Winograd Scheme:

Joan made sure to thank Susan for all the help she had GIVEN/RECEIVED. Who is "she"?

One can instantiate this Winograd Scheme to obtain:

Joan made [...] for all the help she had GIVEN. (Answer: Susan.)
Joan made [...] for all the help she had RECEIVED. (Answer: Joan.)

In both sentences there is a problem of correferential pronominal anaphor at the syntactic level. However, there is no

semantic ambiguity as soon as the reader understands the particular sentence, presumably by placing it within a tacit social protocol concerning giving and receiving actions.[13]

Now consider an example combining common knowledge and naive physics:

The large ball crashed right through the table because it was made of STEEL/STYROFOAM. What is "it"?

One is assumed to know the relative density of styrofoam and steel and facts about crashing. Note that in one instantiation the reader is invited to think of a table made of styrofoam (!); despite this no-commonsense scenario, we can still figure out the physical picture.

The sharpest difference between these two examples lies in their relationship with knowledge. The STEEL/STYROFOAM example requires understanding a few facts; the GIVEN/RECEIVED example requires understanding social norms [86,99]. We note that the distinction between social and physical commonsense has been already emphasized by the authors of the WinoGrande solver [98].

As many "non-social" Winograd Schemes actually depend on facts that are not tied to physics, we refer to them as *factual* Winograd Schemes. For instance, note that to solve the following Winograd Scheme one must have two dates of birth and death of two illustrious writers (Ovid and Goethe):

This book introduced Shakespeare to OVID/GOETHE; it was a fine selection of his writing. Who is "his" referring to?

When we look at the original set of Winograd Schemes [25],[14] we see that 36% of the 150 schemes are factual. The other Winograd Schemes have a social context that appears in two flavors, a *causal* one (32%), where the scheme explores a presumed strong expectation that some event causes another event, and an *consequential* one (32%), where the same expectation is rather loose with respect to possible consequences.

The social causal Winograd Schemes in essence exploit our commonsense ability to frame social events in causal terms, as we tend to see and set events causally and to look for a *causal story* behind (sense) data: "Whenever we see patterns, we look for a causal explanation" [82].[15] An example of social causal Winograd Scheme is the one about Joan and Susan presented above. There are even Winograd Schemes that rely on a perception of causality without any actual mechanistic causation. Consider an example:

Jim YELLED AT/COMFORTED Kevin because he was so upset. Who is "he"?

Of course, it must be the case that human subjects can see connections between the statements in a Winograd Scheme (otherwise, how would we possibly solve the scheme?). But on top of this obvious fact, there is the natural human tendency to think about connections in causal terms.

Here is an example of a social consequential Winograd Scheme:

The customer walked into the bank and stabbed one of the tellers. He was immediately taken to the EMERGENCY ROOM/POLICE STATION. Who was taken to the EMERGENCY ROOM/POLICE STATION?

Comparing causal and consequential Winograd Schemes, one can see a difference in punctuation: causal Winograd Schemes often resorts to conjunctions "because" or "but" to help the reader capture meaning, while consequential Winograd Schemes typically have several sentences.

Thus the original set of Winograd Schemes splits in three parts that are approximately identical in size, one of them factual, the other causal, the other consequential (the latter two together consist of the social schemes).

The current Winograd Challenge does not seem able to capture every aspect of commonsense, let alone of intelligence. However, as we argue in the next subsection, many schemes exploit a nontrivial commonsense perception of causality.

[13] Anthropological studies show a wide variety of cultural/social rules throughout different places/people. This Winograd Scheme displays a classic instance of the reciprocity norm presented by Marcel Mauss [66] in which any culture relies on a cycle of actions of giving-receiving-retribution: person_a gives something to person_b, person_b accepts what was given by person_a, finally person_b retributes something else to person_a, with possible repetitions. The prevalence of strong behavioral patterns in social schemes must be one of the reasons why it has been possible, in the end, to attain high accuracy with pattern extraction methods.

[14] There are 150 Winograd Schemes in this set; those schemes lead to 273 sentences that comprise the so-called WSC273 dataset.

[15] Such a claim is cogent within an evolutionary perspective. Our perception works applying a event-event view of causality (any event depends on a previous event). Michotte [69] was the pioneer of the scientific studies on the perception of causality in psychology in the 1960s; his experiments revealed that humans get the impression of a causal interaction between two objects even though there was no interaction between these objects. The story-like experience of the events is stored in our long-term memory as an episodic memory (memory of experiences) as opposed to the semantic memory (memory of facts) a paradigm distinction proposed by Tulving in the 1970s [108]. More recently, studies connect the ability of store story-like chaining of causal events as well as of transferring those experiences to others and the evolution of human species [67].

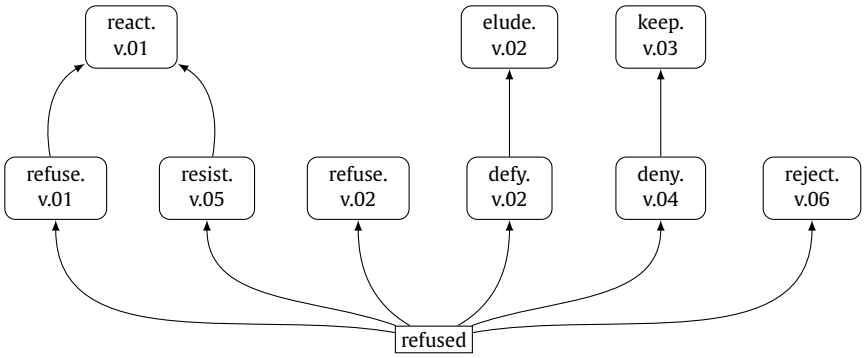


Fig. 2. An example of WordNet's super/sub-ordinate relationships.

6.3. Solving and explaining with Zenograd, a system with self-restraint

We now describe the Zenograd solver, a minimal system that disambiguates causal Winograd Schemes and provides linguistic-based explanations. Our goal was to build a simple system based on linguistic rules and on public knowledge repositories. Actually we restricted ourselves to *one* repository, the WordNet [33], a knowledge graph capturing a vast number of linguistic relations. We now show how causal schemes can be, with such restricted resources,[16] effectively solved and explained (Zenograd reaches 0.896 accuracy on causal schemes).

First, a word on WordNet. It contains a lexical database of semantic resources, including synonymy relations (e.g., car and automobile). Synonyms are grouped into unordered sets referred to as synsets. There are 117000 synsets in WordNet; each one of them can be viewed as a concept. Words with several distinct meanings are associated with as many distinct synsets. The most frequent relation between synsets is the super/sub-ordinate relation (also referred to as hyperonymy, hyponymy, or IS_A relation). For instance, we have that "dog is a hyponym of canine" and "canine is a hypernym of dog." Concerning verbs, adjectives, and adverbs, synsets are organized from the most abstract to the most concrete — a properly that is extensively exploited by Zenograd, as we describe later. Fig. 2 shows a word mapped into a synset in WordNet.

Consider again, now as a running example, the following causal Winograd Scheme:

The city councilmen refused the demonstrators a permit because they FEARED/ADVOCATED violence. Who are "they"?

Zenograd starts by identifying the events (one containing ambiguity, the other not), information about their connection, and possible ways to replace the ambiguous pronoun; either Winograd Schemes are appropriately tagged [25] or a standard tagger is applied [8].

For our running example, Zenograd obtains the following frame-like representation:

Event_1	*A* refused *B* a permit
Event_2	*X* FEARED/ADVOCATED violence.
Connection	Causal
Connecting expression	because
A	The city councilmen
B	The demonstrators
X	They

Zenograd then identifies the kind of connection between *A* and *B*, as this connection raises the expectation of some further caused action. The analysis starts with the event that contains no ambiguous pronoun (in the running example, this is Event_1). Zenograd collects the synsets relating action from *A* to *B* in this event; in the running example, it forms the abstract sentence

$$A \left[\text{synset}_{\text{refused}_i}\right] B \text{ a permit}$$

for each possible synset $\text{synset}_{\text{refused}_i}$ of "refused".

Most schemes center around verbs. We collapsed labels in WordNet to obtain a binary classification for verbs, taking each of them to be either *pragmatic* or *epistemic*. Pragmatic verbs are the ones that make the subject directly deal with the object; for instance, "Joan [...] *thanked* Susan [...]" or "The lawyer *asked* the witness a question [...]." On the other hand,

[16] Of course, the WordNet is not a restricted resource by itself. It has already been built using many hours of human effort and now knowledge embedded in the WordNet can be used at very low cost. It is the Zenograd system itself that requires modest resources.

epistemic verbs do not impose any effect in the object; for instance, emotional verbs: "Pete *envies* Martin [...]" or "They *feared* violence." Whenever necessary, Zenograd retrieves the classification between pragmatic and epistemic from WordNet by inheriting through super-ordinate relations (hypernyms). In going up the hierarchy of synsets, we avoid uninformative concepts such as thing.n.01 or substance.n.01 by blocking some paths — we marked them manually after analyzing the WordNet graph.

Winograd Schemes also require reasoning about adjectives, adverbs, verbs. Each word is associated with a binary *valence* that can be 0 or 1. For verbs, we get 0 for epistemic ones and 1 for pragmatic ones; whenever necessary, Zenograd goes up in the WordNet hierarchy to classify a verb using labels in WordNet. For instance, in Fig. 2 the verb react.v.01 is classified as pragmatic, and therefore gets valence 1. The children of this node, refuse.v.01 and resist.v.05, inherit valence 1. In our running example, we have that "fear" (associated with Event_2) is epistemic and gets valence 0, while "advocate" is a pragmatic verb that gets valence 1. An analogue valence is attached to adjectives and adverbs: "good" gets valence 1 and "bad" gets valence 0. Because these labels are not directly found in WordNet, we manually labeled a couple of hundred high level concepts and all the other adjectives and adverbs get labels from them by traveling up the WordNet graph.

Valence is a working concept that we have developed for Zenograd. The core of the idea is derived from Lakoff and Johnson watershed psychological work on conceptual metaphors as portrayed in their book *Metaphors We Live By* [60]. We do not have space to dig deeper into their ideas; we simply note that there exists no actual implementation of their proposals, as we have done, in a classification task. In any case, conceptual metaphors have been employed by researchers in natural language understanding [93]. Lakoff's key point suffice to understand our pragmatic purpose here: human beings organize their speech according to some binary bodily metaphors, such as up / down, left / right, inside / outside. This is clear when we deal with adjectives and adverbs. Take for instance the association we have with "good" and "up" (as in *thumbs up*) and bad with "down" (as in *thumbs down*). Such spatial divisions are not neutral, and that is why we adopted the term "valence". One side of the divided space is to be less valued or neutrally valued when compared to the other side. Because we wanted to keep our model with minimal size, we combined the neutral and negative values of adjectives and adverbs into label 0, leaving the positive valence with label 1.

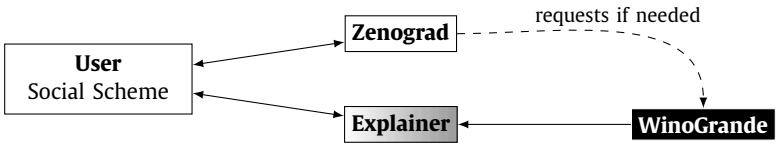
When it comes to the verbs, the definition of valence is somewhat different because there is no salient valuation as with adjective and adverbs. However, we can say that verbs have a "spatial direction", to insist with the spatial and bodily metaphor. What we refer to as pragmatic is related to action and more specifically to *change in directed action*. We wanted also to cover social actions in the sense of actions directed to other entities [114] (that may be a passive attitudes such as believing, fearing, hoping). Passive attitudes have as their objects non material entities; hence we called them epistemic. Pragmatic utterances then get label 1 and epistemic utterances get label 0. However, we had to take into account that, in some social actions, some verbs might be outer directed or inner directed. For instance, *selling* from the point of view of the seller might be valued as 1, where buying from the buyers perspective might be valued as 0. Valence is then a tool to eliminate ambiguities concerning the direction of actions. The rationale behind that is simple: if two actors are with the same orientation, with the same valence, at the same moment, they are actually acting together, so they are a collective entity. Contemporary work on social ontology explore this sort of relationship with the concept of *shared agency* [109,110].

Zenograd attaches valences to all relevant words in an input Winograd Scheme and then reasons about the possible linguistic arrangements around valences. In short: if A did something to B and this action has valence 1, then it is expected that either B reacts to A with a action of valence 1 or that A expresses her beliefs, justifications, or feelings about the situation. This is exactly the structure of our running example. Conversely, if A had a belief or a feeling towards B, it is expected that some property attached to B gets valence 1. This is actually the structure behind the sentence "Pete envies Martin although he is very successful."

With this rather concise approach, where the complexity lies in the employed linguistic knowledge and not in the solver itself, Zenograd does an excellent job at causal Winograd Schemes. In fact Zenograd can get as input any social Winograd Scheme, as its first decision is to determine whether the input scheme is a causal or consequential one. For the 32% of Winograd Schemes in WSC273 [25] that are causal, Zenograd obtains 0.896 accuracy (and note that these are schemes that Zenograd itself determined to be causal from the set of social Winograd Schemes). So, for causal Winograd Schemes, we get high accuracy and linguistic structure that can lead to explanations of the sort discussed previously.

As for consequential Winograd Schemes, they fall outside of Zenograd's abilities: for the 36% of Winograd Schemes in WSC273 that Zenograd itself determined to be consequential, Zenograd's accuracy is just 0.5. Much more subtle reasoning is needed to solve consequential Winograd Schemes. It certainly does not seem that a compact KRR strategy can perform at the accuracy level of data-driven solvers; more elaborate schemes should be explored in future work.

It is here that our argument comes to a close, as we would like to suggest that the best current way to handle the Explaining Winograd Challenge (restricted to social schemes) is to adopt a "parallel" combination as indicated in the diagram below. That is, a social Winograd scheme can be processed both by Zenograd and a top data-driven solver such as the WinoGrande solver, but the former directly offers an explanation, while the latter would, when so asked by the former, provide a solution and resort to an explainer. Horizontal strategies alluded to earlier can then be used by the explainer to fill in the explanation that the data-driven solver cannot provide. Zenograd is brittle because it cannot deal with every input; the WinoGrande solver is brittle because it cannot manage the new challenges imposed by explanations; both of them can collaborate positively.



Of course one might look for a series combination where a single data-driven solver is explained, perhaps linked to another system that learns to explain. Before we go and spend millions paying people to produce explanations so that machines can learn from them, we might just realize how much power can be extracted from a bit of reasoning and a large store of knowledge. A parallel collaboration seems a better strategy.

To conclude, note that we have not addressed factual Winograd Schemes at all in Zenograd, because factual schemes require information from additional knowledge bases. At this stage we wanted to avoid a discussion on what sort of additional knowledge one can bring into the solver beyond purely linguistic conventions, so we leave the solution/explanation of factual Winograd Schemes to future work.

7. Conclusion

We have already traveled through a long maze of ideas, so it is better to keep this conclusion short. Our main point is simple: KRR can enhance machine learning both by providing a theoretical perspective and associated machinery, and by providing economical ways to improve practical solutions. We have emphasized a particular mix of KRR and data-driven machine learning, where selected symbolic modules are brought in to enhance data-driven techniques.

To summarize, in this paper we looked both at theoretical tools and at practical procedures:

- Regarding the former, we argued in Section 4 that various KRR tools support analysis and design of modeling languages to be used as targets of data-driven methods.
- Regarding the latter, we have selected a few practical settings where explanation generation is supported by symbolic tools as discussed in Sections 5 and 6.

Given the stellar performance of data-driven techniques, one must be wise in mining for those opportunities where KRR can make a cost-effective difference when compared to massive data processing. To recap a few key points made in the paper: sometimes we can insert background knowledge into a model, perhaps to regularize it, or to force a desirable inductive bias, or to prevent discriminatory behavior; sometimes it will be profitable to have symbolic modules to explain or to debug a working system; sometimes it will be useful to transfer learned facts and rules. At some future point we will perhaps teach machines not just with data but also by symbolic explanation, as we routinely do in school. Indeed, Turing speculated in his 1950 piece whether the path to AI would be to have a machine “simulate the child’s” mind in learning, possibly by taking “orders given in some language, e.g., symbolic language” [111].

It is a popular saying, backed by human experience, that knowledge is power. Today we realize, more than ever, that data are power as well. Perhaps their combination will lead to machines with real (super?) power.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The first author has been partially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), grant 312180/2018-7. The second author has been supported by the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), grant 2018/09681-4.

The work was also supported by the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), grants 2016/18841-0 and 2019/07665-4, and also by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) - finance code 001.

References

[1] M. Alshammari, O. Nasraoui, S. Sanders, Mining semantic knowledge graphs to add explainability to black box recommender systems, IEEE Access 7 (2019) 110563–110579.

[2] Robert Audi, Epistemology: A Contemporary Introduction to the Theory of Knowledge, Rutledge, 2010.

[3] S. Bader, P. Hitzler, Dimensions of neural-symbolic integration — a structured survey, in: We Will Show Them! Essays in Honour of Dov Gabbay, College Publications, 2005, pp. 167–194.

[4] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, Razvan Pascanu, Relational inductive biases, deep learning, and graph networks, Technical report, arXiv:1806.01261, 2018.
[5] Paul Beame, Guy Van den Broeck, Eric Gribkoff, Dan Suciu, Symmetric weighted first-order model counting, in: ACM Symposium on Principles of Database Systems, PODS, 2015, pp. 313–328.
[6] D. Bender, Establishing a human baseline for the Winograd schema challenge, in: Modern AI and Cognitive Science Conference, 2015, pp. 39–45.
[7] Jay Bennett, Riddle of the week #13: the spies with stamps on their heads, Pop. Mech. (January 2017).
[8] Steven Bird, Edward Loper, Ewan Klein, Natural Language Processing with Python, O'Reilly, 2009.
[9] David M. Blei, Andrew Y. Ng, Michael I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[10] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, Jamie Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: ACM SIGMOD International Conference on Management of Data, 2008, pp. 1247–1250.
[11] Ronald J. Brachman, Hector J. Levesque, Knowledge Representation and Reasoning, Morgan Kaufmann, 2004.
[12] Antony Browne, Ron Sun, Connectionist inference models, Neural Netw. 14 (10) (2001) 1331–1355.
[13] Brune G. Buchanan, Reid G. Smith, Fundamentals of expert systems, Annu. Rev. Comput. Sci. 3 (1988) 23–58.
[14] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, DL-Lite: tractable description logics for ontologies, in: AAAI, 2005, pp. 602–607.
[15] Timothy Chappell, Reading Plato's Theaetetus, Hackett Publishing, 2004.
[16] Fabio G. Cozman, Languages for probabilistic modeling over structured and relational domains, in: Pierre Marquis, Odile Papini, Henri Prade (Eds.), A Guided Tour of Artificial Intelligence Research, vol. 2, Springer, 2020, pp. 247–283, chapter 9.
[17] Fabio G. Cozman, Denis D. Mauá, The complexity of Bayesian networks specified by propositional and relational languages, Artif. Intell. 262 (2018) 96–141.
[18] Fabio G. Cozman, Denis D. Mauá, The finite model theory of Bayesian network specifications: descriptive complexity and zero/one laws, Int. J. Approx. Reason. 110 (2019) 107–126.
[19] Paulo C.G. da Costa, Kathryn B. Laskey, Of Klingons and starships: Bayesian logic for the 23rd century, in: Conference on Uncertainty in Artificial Intelligence, 2005.
[20] Arthur d'Avila Garcez, Luís Lamb, Neurosymbolic AI: the 3rd wave, Technical report, arXiv:2012.0587, 2020.
[21] Arthur S. d'Avila Garcez, Krysia Broda, Dov M. Gabbay, Neural-Symbolic Learning Systems: Foundations and Applications, Springer, 2002.
[22] Arthur S. d'Avila Garcez, Luís C. Lamb, Dov M. Gabbay, Neural-Symbolic Cognitive Reasoning, Springer, 2009.
[23] Artur d'Avila Garcez, Marco Gori, Luís C. Lamb, Luciano Serafini, Michael Spranger, Son N. Tran, Neural-symbolic computing: an effective methodology for principled integration of machine learning and reasoning, IfCoLog J. Log. Appl. 6 (4) (June 2019).
[24] Ernest Davis, Representations of Commonsense Knowledge, Morgan Kaufmann, 1990.
[25] Ernest Davis, Collection of Winograd schemas, Available at site https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WSCollection.html, 2018.
[26] Luc De Raedt, Logical and Relational Learning, Springer, 2008.
[27] Luc De Raedt, Paolo Frasconi, Kristian Kersting, Stephen Muggleton, Probabilistic Inductive Logic Programming, Springer, 2008.
[28] Luc De Raedt, Kristian Kersting, Probabilistic inductive logic programming, in: International Conference on Algorithmic Learning Theory, 2004, pp. 19–36.
[29] Pedro Domingos, Daniel Lowd, Markov Logic: An Interface Layer for Artificial Intelligence, Morgan and Claypool, 2009.
[30] Saso Dzeroski, Multi-relational data mining: an introduction, ACM SIGKDD Explor. Newsl. 5 (1) (2003) 1–16.
[31] Saso Dzeroski, Nada Lavrac, Relational Data Mining, Springer, Berlin, 2001.
[32] Ronald Fagin, Probabilities on finite models, J. Symb. Log. 41 (1) (1976) 50–58.
[33] Christiane Fellbaum, WordNet: An Electronic Lexical Database, The MIT Press, 1998.
[34] Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shrerionov, Bernd Gutmann, Gerda Janssens, Luc De Raedt, Inference and learning in probabilistic logic programs using weighted Boolean formulas, Theory Pract. Log. Program. 15 (3) (2014) 358–401.
[35] Nir Friedman, Lise Getoor, Daphne Koller, A. Pfeffer, Learning probabilistic relational models, in: International Joint Conference on Artificial Intelligence, 1999, pp. 1300–1309.
[36] Matt Gardner, Tom Mitchell, Efficient and expressive knowledge base completion using subgraph feature extraction, in: Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1488–1498.
[37] Matt Gardner, Partha Talukdar, Tom Mitchell, Combining vector space embeddings with symbolic logical inference over open-domain text, in: Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches — Papers from the 2015 AAAI Spring Symposium, 2015, pp. 61–65.
[38] Lise Getoor, Ben Taskar, Introduction to Statistical Relational Learning, MIT Press, 2007.
[39] Walter R. Gilks, Andrew Thomas, David Spiegelhalter, A language and program for complex Bayesian modelling, Statistician 43 (1993) 169–178.
[40] Andrew D. Gordon, Thomas A. Henzinger, Aditya V. Nori, Sriram K. Rajmani, Probabilistic programming, in: Future of Software Engineering, ACM, 2014, pp. 167–181.
[41] Erich Grädel, Finite model theory and descriptive complexity, in: Finite Model Theory and Its Applications, Springer, 2007, pp. 125–229.
[42] Erich E. Grädel, Phokion G. Kolaitis, Leonid Libkin, Maarten Marx, Joel Spencer, Moshe Y. Vardi, Yde Venema, Scott Weinstein, Finite Model Theory and Its Applications, Springer, 2007.
[43] David Gunning, Explainable artificial intelligence (XAI), Technical Report DARPA-BAA-16-53, Defense Advanced Research Projects Agency, 2016.
[44] Arthur Colombini Gusmão, Alvaro Henrique Chaim Correia, Glauber De Bona, Fabio Gagliardi Cozman, Interpreting embedding models of knowledge bases: a pedagogical approach, in: ICML Workshop on Human Interpretability in Machine Learning (WHI 2018), 2018, pp. 79–86.
[45] Alon Halevy, Peter Norvig, Fernando Pereira, The unreasonable effectiveness of data, IEEE Intell. Syst. 24 (2009) 8–12.
[46] Barbara Hammer, Pascal Hitzler, Perspectives of Neural-Symbolic Integration, Springer, 2007.
[47] Melanie Hilario, An overview of strategies for neurosymbolic integration, in: Ron Sun, Frederic Alexandre (Eds.), Connectionist-Symbolic Integration, 1997, pp. 13–36.
[48] D.J. Hilton, Conversational processes and causal explanation, Psychol. Bull. 107 (1990) 65–81.
[49] Manfred Jaeger, Relational Bayesian networks, in: Dan Geiger, Prakash Pundalik Shenoy (Eds.), Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, California, 1997, pp. 266–273.
[50] Manfred Jaeger, Model-theoretic expressivity analysis, in: L. De Raedt, P. Frasconi, K. Kersting, S. Muggleton (Eds.), Probabilistic Inductive Logic Programming, Springer, 2008, pp. 325–339.
[51] D. Kahneman, A. Tversky, The simulation heuristic, in: D. Kahneman, P. Slovic, A. Tversky (Eds.), Judgment Under Uncertainty: Heuristics and Biases, Cambridge University Press, 1998.
[52] Daniel Kahneman, Thinking, Fast and Slow, Farrar, Straus and Giroux, 2011.
[53] Daniel Kahneman, Varieties of counterfactual thinking, in: N. Roese, J. Olson (Eds.), What Might Have Been: the Social Psychology of Counterfactual Thinking, Psychology Press, New York, 2014, pp. 375–396.

[54] Gabriele Kern-Isbener, Christoph Beierle, Marc Finthammer, Matthias Thimm, Comparing and evaluating approaches to probabilistic reasoning: theory, implementation, and applications, in: Transactions on Large-Scale Data- and Knowledge-Centered Systems VI, 2012, pp. 31–75.
[55] Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, Thomas Lukasiewicz, A surprisingly robust trick for the Winograd scheme challenge, in: Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4837–4842.
[56] Vid Kocijan, Thomas Lukasiewicz, Ernest Davis, Gary Marcus, Leora Morgenstern, A review of Winograd Schema challenge datasets and approaches, Technical report arXiv:2004.13831, 2020.
[57] Daphne Koller, Probabilistic relational models, in: Inductive Logic Programming, in: LNCS, vol. 1634, Springer, 1999, pp. 3–13.
[58] Daphne Koller, Nir Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.
[59] Daphne Koller, Avi Pfeffer, Object-oriented Bayesian networks, in: Conference on Uncertainty in Artificial Intelligence, 1997, pp. 302–313.
[60] George Lakoff, Mark Johnson, Metaphors We Live by, The University of Chicago Press, 1980.
[61] Luís C. Lamb, Artur d'Avila Garcez, Marco Gori, Marcelo O.R. Prates, Pedro H.C. Avelar, Moshe Y. Vardi, Graph neural networks meet neural-symbolic computing: a survey and perspective, in: International Joint Conference on Artificial Intelligence, 2020, pp. 4877–4884.
[62] Hector J. Levesque, Common Sense, the Turing Test, and the Quest for Real AI, MIT Press, 2017.
[63] Hector J. Levesque, E. Davis, L. Morgenstern, The Winograd schema challenge, in: International Conference on Principles of Knowledge Representation and Reasoning, 2012, pp. 552–561.
[64] Quan Liu, Hui Jiang, Zhen-Hua Ling, Xiaodan Zhu, Si Wei, Yu Hu, Commonsense knowledge enhanced embeddings for solving pronoun disambiguation problems in Winograd schemes challenge, Technical report arXiv:1611.04146, 2016.
[65] Suzanne Mahoney, K.B. Laskey, Network engineering for complex belief networks, in: Conference on Uncertainty in Artificial Intelligence, 1996.
[66] Marcel Mauss, The Gift: Forms and Functions of Exchange in Archaic Societies, Cohen & West, 1966.
[67] Glen McBride, Storytelling, behavior planning, and language evolution in context, Front. Psychol. 15 (1131) (2014) 1–11.
[68] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (1943) 115–133.
[69] Albert Michotte, The Perception of Causality, Basic Books, 1963.
[70] Tim Miller, Explanation in artificial intelligence: insights from the social sciences, Artif. Intell. J. 267 (2019) 1–38.
[71] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, J. Welling, Never-ending learning, in: AAAI Conference on Artificial Intelligence, 2015, pp. 2302–2310.
[72] Tom Mitchell, Machine Learning, McGraw-Hill, 1997.
[73] Christoph Molnar, Interpretable Machine Learning, Lulu, 2019, https://christophm.github.io/interpretable-ml-book/.
[74] Stephen Muggleton, Inductive logic programming, New Gener. Comput. 8 (1991) 295–318.
[75] Stephen Muggleton, L. De Raedt, Inductive logic programming: theory and methods, J. Log. Program. 20 (1994) 629–679.
[76] Stephen H. Muggleton, Ute Schmid, Christina Zeller, Alireza Tamaddoni-Nezhad, Tarek Besold, Ultra-strong machine learning: comprehensibility of programs learned with ILP, Mach. Learn. 107 (2018) 1119–1140.
[77] Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
[78] C. Musto, F. Narducci, P. Lops, M. de Gemmis, G. Semeraro, Linked open data-based explanations for transparent recommender systems, Int. J. Hum.-Comput. Stud. 121 (2019) 93–107.
[79] Maximilian Nickel, Kevin Murphy, Volker Tresp, Evgeniy Gabrilovich, A review of relational machine learning for knowledge graphs, Proc. IEEE 104 (1) (2015) 11–33.
[80] Nils J. Nilsson, The Quest for Artificial Intelligence, Cambridge University Press, 2010.
[81] Judea Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, California, 1988.
[82] Judea Pearl, Causality: Models, Reasoning, and Inference, 2nd edition, Cambridge University Press, Cambridge, United Kingdom, 2009.
[83] Jorge Pérez, Javier Marinković, Pablo Barceló, On the Turing completeness of modern neural network architectures, in: International Conference on Learning Representations, 2019.
[84] Gustavo Padilha Polleti, Hugo Neri Munhoz, Fabio Gagliardi Cozman, Explanations within conversational recommendation systems: improving coverage through knowledge graph embedding, in: AAAI Workshop on Interactive and Conversational Recommendation Systems, 2020, pp. 1–8.
[85] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, Language models are unsupervised multitask learners, Technical Report 8, OpenAI Blog, 2019.
[86] Joseph Raz, Practical Reasoning and Norms, Hutchinson, 1975.
[87] Raymond Reiter, Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems, MIT Press, 2001.
[88] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, Why should I trust you?: Explaining the predictions of any classifier, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[89] Francesco Ricci, Lion Rokach, Bracha Shapira, Paul B. Kantor, Recommender Systems Handbook, Springer, 2011.
[90] E. Rich, K. Knight, Artificial Intelligence, McGraw-Hill, New York, 1991.
[91] Matthew Richardson, Pedro Domingos, Markov logic networks, Mach. Learn. 62 (1–2) (2006) 107–136.
[92] R. Riegel, A.G. Gray, F.P.S. Luus, N. Khan, N. Makondo, I.Y. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma, S. Ikbal, H. Karanam, S. Neelam, A. Likhyani, S.K. Srivastava, Logical neural networks, Technical report arXiv:2006.13155, 2020.
[93] Zachary Rosen, Computationally constructed concepts: a machine learning approach to metaphor interpretation using usage-based construction grammatical cues, in: Proceedings of the Workshop on Figurative Language Processing, ACL, 2018, pp. 102–109.
[94] Roth Dan, On the hardness of approximate reasoning, Artif. Intell. 82 (1–2) (1996) 273–302.
[95] Yu-Ping Ruan, Xiaodan Zhu, Zhen-Hua Ling, Zhan Shi, Quan Liu, Si Wei, Exploring unsupervised pretraining and sentence structure modelling for Winograd schemes challenge, Technical report arXiv:1904.09705, 2019.
[96] Andrey Ruschel, Arthur Colombini Gusmão, Gustavo Padilha Polleti, Fabio Gagliardi Cozman, Explaining completions produced by embeddings of knowledge graphs, in: European Conference on Quantitative and Symbolic Approaches to Reasoning with Uncertainty, 2019, pp. 324–335.
[97] S. Russell, J. Binder, D. Koller, K. Kanazawa, Local learning in probabilistic networks with hidden variables, in: Fourteenth International Joint Conference on Artificial Intelligence, 1995, pp. 1146–1152.
[98] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, Yejin Choi, Winogrande: an adversarial Winograd schema challenge at scale, Technical report arXiv:1907.10641, 2019.
[99] T. Scanlon, Being Realistic About Reasons, Oxford University Press, 2014.
[100] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, Gabriele Monfardini, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2009) 61–80.
[101] Roger C. Schank, Explanation Patterns: Understanding Mechanically and Creatively, Psychology Press, 1986.
[102] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, Int. J. Comput. Vis. 128 (2020) 336–359.
[103] Luciano Serafini, Arthur S. d'Avila Garcez, Learning and reasoning with Logic Tensor Networks, in: Conference of the Italian Association for Artificial Intelligence, 2016, pp. 334–348.

[104] E.H. Shortliffe, B.G. Buchanan, Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, Reading, Mass, 1984.
[105] Emma Strubell, Ananya Ganesh, Andrew McCallum, Energy and policy considerations for deep learning in NLP, in: Association of Computational Linguistics, 2019.
[106] N. Tintarev, J. Masthoff, A survey of explanations in recommender systems, in: IEEE International Conference on Data Engineering Workshop, 2007, pp. 801–810.
[107] Trieu H. Trinh, Quoc V. Le, A simple method for commonsense reasoning, Technical report arXiv:1806.02847, 2018.
[108] Endel Tulving, Episodic and semantic memory, in: E. Tulving, W. Donaldson (Eds.), Organization of Memory, Academic Press, 1972, pp. 381–403.
[109] Raimo Tuomela, The We-mode and the I-mode, in: F. Schmitt (Ed.), Socializing Metaphysics: the Nature of Social Reality, Rowman & Littlefield, 2003, pp. 93–127.
[110] Raimo Tuomela, The Philosophy of Sociality: The Shared Point of View, Oxford University Press, 2007.
[111] A.M. Turing, Computing machinery and intelligence, Mind LIX (1950) 433–460.
[112] Guy Van den Broeck, Dan Suciu, Query processing on probabilistic data: a survey, Found. Trends® Databases (2017) 197–341.
[113] Frank van Harmelen, Vladimir Lifschitz, Bruce Porter, Handbook of Knowledge Representation, Elsevier, 2007.
[114] Max Weber, Economy and Society; an Outline of Interpretive Sociology, Bedminster Press, 1968.
[115] M.P. Wellman, J.S. Breese, R.P. Goldman, From knowledge bases to decision models, Knowl. Eng. Rev. 7 (1) (1992) 35–53.
[116] Terry Winograd, Natural Language Understanding, Academic Press, 1972.
[117] Keyulu Xu, Weihua Hu, Jure Leskovec, Stefanie Jegelka, How powerful are graph neural networks?, in: International Conference on Learning Representations, 2019, pp. 1–17.
[118] W. Zhang, B. Paudel, W. Zhang, A. Bernstein, H. Chen, Interaction embeddings for prediction and explanation in knowledge graphs, in: ACM International Conference on Web Search and Data Mining, 2019, pp. 96–104.