Modelling Symbolic Knowledge using Neural Representations

Steven Schockaert and Víctor Gutiérrez-Basulto

Cardiff University, UK

Abstract. Symbolic reasoning and deep learning are two fundamentally different approaches to building AI systems, with complementary strengths and weaknesses. Despite their clear differences, however, the line between these two approaches is increasingly blurry. For instance, the neural language models which are popular in Natural Language Processing are increasingly playing the role of knowledge bases, while neural network learning strategies are being used to learn symbolic knowledge, and to develop strategies for reasoning more flexibly with such knowledge. This blurring of the boundary between symbolic and neural methods offers significant opportunities for developing systems that can combine the flexibility and inductive capabilities of neural networks with the transparency and systematic reasoning abilities of symbolic frameworks. At the same time, there are still many open questions around how such a combination can best be achieved. This paper presents an overview of recent work on the relationship between symbolic knowledge and neural representations, with a focus on the use of neural networks, and vector representations more generally, for encoding knowledge.

1 Introduction

Artificial Intelligence (AI) is built on two fundamentally different traditions, both of which go back to the early days of the field. The first tradition is focused on formalising human reasoning using symbolic representations. This tradition has developed into the Knowledge Representation and Reasoning (KRR) sub-field. The second tradition is focused on learning from examples. This tradition has developed into the Machine Learning (ML) sub-field. These two different traditions have complementary strengths and weaknesses. Due to the use of symbolic representations, KRR systems are explainable, often come with provable guarantees (e.g. on correctness or fairness) and they can readily exploit input from human experts. Moreover, due to their use of systematic reasoning processes, KRR systems are able to derive conclusions that require combining numerous pieces of knowledge in intricate ways. However, symbolic reasoning is too rigid for many applications, where predictions may need to be made about new situations that are not yet covered in a given knowledge base. On the other hand, ML systems often require little human input, but lack explainability, usually come without guarantees, and tend to struggle in applications where systematic

reasoning is needed [1–3]. Accordingly, there is a growing realisation that future AI systems will need to rely on an integration of ideas from ML and from KRR.

The integration of symbolic reasoning with neural models already has a long tradition within the context of neuro-symbolic AI [4, 5]. However, our main focus in this overview is not on the integration of symbolic reasoning with neural network learning, but on the ability of neural network models, and vector space encodings more generally, to play the role of knowledge bases. First, in Section 2, we focus on the use of neural models for capturing knowledge graphs (i.e. sets of dyadic relational facts). Knowledge graphs play an important role in research fields such as Natural Language Processing, Recommendation and Machine Learning, essentially giving AI system access to factual world knowledge. The interest in studying the relationship between neural models and knowledge graphs is two-fold. On the one hand, learning vector representations of knowledge graphs makes it easier to use these resources in downstream tasks. On the other hand, existing pre-trained neural language models, trained from large text collections, implicitly capture a lot of the information that is stored in open-domain knowledge graphs. Neural models can thus also play an important role in constructing or extending knowledge graphs. In Section 3, we then look at the ability of neural models to capture rules, e.g. the kind of knowledge that would normally be encoded in ontologies. Studying this ability is important because it can suggest mechanisms to combine traditional strategies for rule-based reasoning with neural network learning. Moreover, large pre-trained neural language models can also be used as a source of ontological knowledge, at least to a certain extent. Finally, in Section 4 we look at cases where neural models and symbolic knowledge are jointly needed. This includes, for instance, the use of existing rule bases, along with traditional labelled examples, for training neural models. Moreover, symbolic representations are also used for querying neural representations. As a final example, we look at mechanisms to exploit neural representations for making symbolic reasoning more flexible or robust.

2 Encoding Knowledge Graphs

A knowledge graph (KG) is a set of triples of the form $(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, with $\mathcal{E}$ a set of entities and $\mathcal{R}$ a set of relations. A triple (h, r, t) intuitively expresses that the head entity h and tail entity t are in relation r. For instance, (*Cardiff*, *capital-of*, *Wales*) asserts that Cardiff is the capital of Wales. KGs are among the most popular frameworks for encoding factual knowledge. Open-domain KGs such as Wikidata [6], YAGO [7] and DBpedia [8], can be seen as providing a structured counterpart to Wikipedia. Such KGs are commonly used as a source of factual encyclopedic information about the world, for instance to enrich neural network models for Natural Language Processing (NLP) [9]. Commonsense KGs such as ConceptNet [10] and ATOMIC [11] are similarly used as a source of knowledge that may otherwise be difficult to obtain. Furthermore, a large number of domain-specific KGs have been developed, for instance covering the needs of a specific business. We refer to [12, 13] for a comprehensive overview

about knowledge graphs. Here we focus on neural representations of KGs. The aim of using neural representations is to generalise from the facts that are explicitly asserted in a given KG and to make it easier to take advantage of KGs in downstream tasks. In Section 2.1, we first discuss KG embedding (KGE) methods, i.e. strategies for learning vector representations of entities and relations that capture the knowledge encoded in a given KG. Such methods have seen a lot of attention from the research community throughout the last decade, having the advantage of being conceptually elegant and computationally efficient. In Section 2.2 we then discuss the use of Contextualised Language models (CLMs) such as BERT [14] for capturing knowledge graph triples.

2.1 Knowledge Graph Embeddings

Let a knowledge graph $K \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ be given. The aim of knowledge graph embedding (KGE) methods is to learn (i) a vector representation $\mathbf{e}$ for each entity e from $\mathcal{E}$, and (ii) the parameters of a scoring function $f_r : \mathcal{E} \times \mathcal{E} \to \mathbb{R}$ for each relation $r \in \mathcal{R}$, such that $f_r(\mathbf{h}, \mathbf{t})$ reflects the plausibility of the triple (h, r, t). The main focus is usually on the task of *link prediction*, i.e. given a head entity h and relation r, predicting the most likely tail entity t that makes (h, r, t) a valid triple. Embeddings are typically real-valued, i.e. $\mathbf{e} \in \mathbb{R}^n$, but other choices have been considered as well, including complex embeddings [15–17], hyperbolic embeddings [18] and hypercomplex embeddings [19]. In most models, the scoring function f_r is parameterised by a vector $\mathbf{r}$ of the same dimensionality as the entity vectors. For example, in the seminal TransE model [20], we have:

$$f_r(h, t) = -d(\mathbf{h} + \mathbf{r}, \mathbf{t})$$

where d is either the Euclidean or Manhattan distance. In other words, relations are viewed as vector translations, and (h, r, t) is considered plausible if applying the translation for r to $\mathbf{h}$ yields a vector that is similar to $\mathbf{t}$. As another popular example, in DistMult [21], the scoring function is defined as follows:

$$f_r(h, t) = \mathbf{h} \odot \mathbf{r} \odot \mathbf{t}$$

where $\odot$ denotes the component-wise product of vectors. To learn the entity vectors and the scoring functions f_r, several loss functions have been considered, which are typically based on the idea that $f_r(h, t)$ should be higher than $f_r(h, t')$ whenever $(h, r, t) \in K$ and $(h, r, t') \notin K$. An important lesson from research on KGE is that the performance of different methods often crucially depends on the chosen loss function, the type of regularisation that is used, how the negative examples (h, r, t') are chosen, and hyper-parameter tuning [22]. This has complicated the empirical comparison of different KGE models, especially given that these models are typically only evaluated on a small set of benchmarks.

Leaving empirical considerations aside, an important question is whether KGE models have any theoretical limitations on the kinds of KGs they can encode. In other words, is it always possible to find entity vectors and scoring functions such that the triples (h, r, t) which are predicted to be valid by the

KGE model are exactly those that are contained in a given KG? Formally, a KGE model is called fully expressive [23] if for any knowledge graph K, we can find entity vectors and parameters for the scoring functions such that $f_r(h,t) > \gamma$ if $(h,r,t) \in K$ and $f_r(h,t) < \gamma$ otherwise, for some constant $\gamma \in \mathbb{R}$. In other words, a fully expressive model is capable of capturing any knowledge graph configuration. It turns out that basic translation based methods such as TransE are not fully expressive (see [23] for details). However, many other methods have been found to be fully expressive [23, 15], provided that vectors of sufficiently high dimensionality are used. This also includes BoxE [24], which is translation based but avoids the limitations of other translation based models by using a region based representation.

2.2 Contextualised Language Models as Knowledge Bases

In recent years, the state-of-the-art in NLP has been based on large pre-trained neural language models (LMs) such as BERT [14]. These LMs are essentially deep neural networks that have been pre-trained on large text collections using different forms of self-supervision. The most common pre-training strategy is based on masked language modelling, where the model is trained to predict words from a given input sentence or paragraph that have been masked. Despite the lack of any explicit supervision signal, the resulting LMs have been found to capture a wealth of syntactic and semantic knowledge [25]. Interestingly, these models also capture a lot of factual world knowledge. For instance, [26] found that presenting BERT with an input such as "*Dante was born in <mask>*" leads to the correct prediction (Florence). In fact, it turns out that pre-trained LMs can be used to answer a wide array of questions, without being given acccess to any external knowledge or corpus [27]. Rather than using KG embeddings to provide NLP models with access to knowledge about the world, the focus in recent years has thus shifted towards (i) analysing to what extent pre-trained LMs already capture such knowledge and (ii) fine-tuning LMs to inject additional knowledge. LMs thus provide a neural encoding of factual world knowledge, although the mechanism by which such knowledge is encoded is unclear. Recent work [28] has suggested that the feedforward layers of these LMs contain neurons that encode specific facts. This insight was used in [29] to devise a strategy to update the knowledge encoded by an LM, for instance when a given fact has become outdated. Some approaches have been suggested for incorporating KGs when training LMs [30], which provides more control about the kind of knowledge that is captured by the LM. Other methods focus on using KGs to reason about the output of LMs [31]. LMs have also been used to aid in the task of KG completion. For instance, [32] designs a scoring function for KG triples, which uses BERT for encoding entity descriptions. Most notably, LMs have been used for link prediction in commonsense KGs such as ConceptNet and ATOMIC. The challenge with such KGs is that entities often correspond to phrases, which may only appear in a single triple. The graph structure is thus too sparse for traditional KG completion methods to be successful. Instead, [33] proposes a model in which a contextualised language model is fine-tuned on

KG triples. They show that, after this fine-tuning step, the LM can be used to generate meaningful new triples. Building on this work, [34] shows that focused commonsense knowledge graphs can be generated on the fly, to provide context for a particular task.

3 Encoding Rules

While knowledge graphs are the *de facto* standard for encoding factual knowledge, more expressive frameworks are needed for encoding generic knowledge. In particular, rules continue to play an important role within AI, and an increasingly important role within NLP. For instance, several competitive strategies for knowledge graph completion based on learned rules have been proposed in recent years [35, 36], having the advantage of being more transparent than KGE methods, and the potential for capturing more expressive types of inference patterns. Our focus in this overview is on the interaction between rules and neural representations. First, we discuss the use of neural networks for simulating rule based reasoning in Section 3.1. Such methods are particularly appealing, because they are able to learn meaningful rules using standard backpropagation, and can be naturally combined with other types of neural models (e.g. to reason about input presented in the form of images). In Section 3.2, we then discuss the view that rules can be modelled in terms of qualitative spatial relationships between region-based representations of concepts and relations. Finally, Section 3.3 discusses the rule reasoning abilities of contextualised language models.

Before moving to the next sections, we start by briefly introducing rules; for more details, see e.g. [37]. An *atom* α is an expression of the form $R(t_1, \ldots, t_n)$, where R is a *predicate symbol* with *arity* n and terms t_i, i.e. *variables* or *constants*. An *rule* σ is an expression of the form

$$B_1 \wedge \ldots \wedge B_n \rightarrow \exists X_1, \ldots, X_j.H, \tag{1}$$

where $B_1, \ldots B_n$and H are atoms and X_m for $1 \leq m \leq j$ are variables. We call $X_1, \ldots, X_j$ the *existential variables of* σ. All other variables occurring in σ are universally quantified. We call a rule with no free variables a *ground rule* and a ground rule with an empty body a *fact*. For example, *fathertOf(john,peter)*, *fathertOf(peter,louise)* are facts expressing that John is the father of Peter and Peter is the father of Louise. As another example, the following rule with a non-empty body and with variables, defines the grandfather relation in terms of the father relation $\mathit{fatherOf}(X, Y) \wedge \mathit{fatherOf}(Y, Z) \rightarrow \mathit{grandfatherOf}(X, Z)$

3.1 Neural Networks for Reasoning with Differentiable Rules

While the discrete nature of classical logic makes it difficult to integrate with neural networks, several authors have explored techniques for encoding differentiable approximations of logical rules. For instance, [38] develops a differentiable approach to rule based reasoning, called Neural Theorem Proving, by replacing

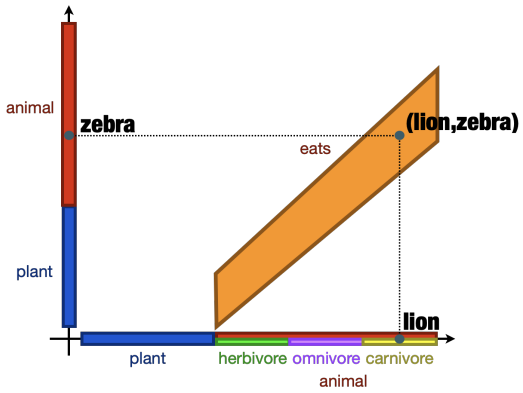

Fig. 1. The binary relation *eats* is defined as a region over the Cartesian product of two conceptual spaces. The spatial configurations capture relational knowledge such as "carnivores only eat animals".

the traditional unification mechanism with a form of soft unification, which is computed based on the dot product between vector representations of the constants and predicates involved. Taking a different approach, Lifted Relational Neural Networks [39] rely on ideas from fuzzy logic to make rules differentiable. In this case, the unification mechanism is the classical one, but truth values of literals and rule bodies are evaluated on a continuous scale. Fuzzy logic connectives are also sometimes used to regularise neural network models based on prior knowledge in the form of rules [40, 41]. DeepProbLog [42] is based on yet another strategy. In this case, reasoning is done using a probabilistic logic program, where a deep neural network is used to estimate the probability of particular literals. Whereas the focus of the aforementioned works was to use neural network learning to discover meaningful rules, in the case of DeepProbLog, the rules themselves are given and the purpose of using neural networks is to allow for more flexible inputs, e.g. making it possible to reason about information presented as images. This strategy has also been instantiated using other logical formalisms, such as answer set programming [43–45]. Some authors have also focused specifically on the use of neural network models for rule induction (rather than for combining rules with neural networks). For instance, [46] presents a differentiable version of inductive logic programming, while [47, 36] propose differentiable models to learn rules for knowledge graph completion.

3.2 Modelling Rules as Spatial relations

The theory of conceptual spaces [48] was proposed by Gärdenfors as an intermediate representation framework, in between neural and symbolic representations. The main idea is that properties are represented as convex regions, while individuals are represented as points. Compared to the usual vector space models, this region-based approach has the advantage that there is a direct correspondence

between spatial relationships in the conceptual space, on the one hand, and symbolic rules, on the other hand. For instance, the fact that individual X satisfies property P, i.e. the fact $P(X)$, corresponds to a situation where the geometric representation of X belongs to the region representing P. Similarly, a rule such as $P(x) \wedge Q(x) \rightarrow R(x)$ corresponds to the situation where the intersection of the regions representing P and Q is included in the region representing R. While conceptual spaces can only capture propositional knowledge, in [49] we showed how relational knowledge can be similarly modelled by representing relations as convex regions over a Cartesian product of conceptual spaces. Figure 1 illustrates this for the binary relation *eats*, which is defined as a convex region over the Cartesian product of two conceptual spaces. The points in this Cartesian product space correspond to pairs of individuals. Relational knowledge is then modelled in terms of inclusions, intersections and projections. For the example illustrated in the figure, among others, the following rules are captured:

$$\textit{carnivore}(x) \wedge \textit{eats}(x, y) \rightarrow \textit{animal}(y)$$
$$\textit{carnivore}(x) \rightarrow \exists y \,.\, \textit{animal}(y) \wedge \textit{eats}(x, y)$$

The framework of conceptual spaces, and their relational extension, seems like a natural choice for settings where neural representations need to be combined with symbolic knowledge. In practice, however, their usage is complicated by the fact that learning regions in high-dimensional spaces is difficult, unless drastically simplifying assumptions are made about the nature of the regions. For example, box embeddings, where entities are represented by hyper-boxes, have been successfully used in a number of contexts [50]. Cones [51, 52] and linear subspaces [53] are also common choices. A typical assumption in conceptual spaces is that regions are defined in terms of the prototypes of the corresponding concepts. Region boundaries may then arise as the cells of a (generalised) Voronoi tessellation of these prototypes [54]. This view is appropriate whenever a *contrast set* [55], i.e. a set of jointly exhaustive and pairwise disjoint concepts is given. In [56], an approach was developed for learning concept representations based on this idea.

3.3 Contextualised Language Models as Rule-Based Reasoners

In Section 2.2, we discussed how large pre-trained language models encode a substantial amount of factual knowledge. The extent to which such language models capture rules is less clear. In [57], some evidence is provided to suggest that LMs are indeed capable of learning some kinds of symbolic knowledge, and can be trained to apply this knowledge, e.g. generalising an observation about a given concept to hyponyms of that concept. In [58], the ability of transformer based LMs to generalise observed facts is analysed in a systematic way, by training an LM from scratch on a synthetic corpus in which various regularities are present. They find that LMs are indeed capable of discovering symbolic rules, and capable of applying such rules for inferring facts not present in the training corpus, although they also identified important limitations. The aforementioned works

mostly focus on one-off rule applications, although some authors have found that LMs can be trained to perform more sophisticated forms of rule based reasoning [59]. Finally, the ability of transformer based LMs to discover and apply rule-like knowledge has also been exploited in the context of knowledge graph completion. Most notably, [60] shows how a fine-tuned BERT model can essentially be used as a rule base for inductive KG completion.

4 Combining Symbolic Knowledge with Embeddings

In the previous section, we discussed how neural networks are able to capture rule-like knowledge to some extent. In many settings, however, symbolic representations also play a central role. For this reason, we now focus on frameworks for *combining* symbolic and neural representations. For instance, symbolic rules can be used to encode knowledge elicited from a domain expert, hence it is of interest to study mechanisms for incorporating symbolic knowledge when training or using neural models, which we discuss in Section 4.1. Symbolic representations are also needed for specifying complex information needs. Recently, approaches have been proposed for evaluating such complex (symbolic) queries against knowledge graph embeddings, and other neural representations (Section 4.2). Finally, in applications where interpretability is a primary concern, symbolic knowledge is clearly preferable over neural representations. However, the brittleness of symbolic reasoning means that purely symbolic methods often break down. Symbolic representations are particularly limiting when it comes to inductive reasoning, which in turn makes it difficult to provide plausible or approximate answers in cases where exact reasoning yields no results. To address such concerns, Section 4.3 discusses methods in which neural representations are used to add inductive capabilities to symbolic frameworks.

4.1 Injecting Knowledge into Neural Models

Rules are commonly used for injecting prior knowledge when training a neural model [61, 62, 41, 63]. The most typical strategy is to approximate the rules using differentiable functions, and to add a term to the loss function which encourages the learned representations to adhere to the rules. Another strategy is to use (heuristic) rules to automatically generate (noisy) labelled training examples [64, 65]. To train a neural model from these noisy labels, the true label is typically modelled as a latent variable, which is inferred by modelling the reliability of the rules, as well as their correlations in some cases. Rather than using symbolic knowledge during training, some approaches also use symbolic knowledge to reason with the output of a neural model. For instance, [66] proposes a model for question answering, which uses a fine-tuned BERT model to generate a vector representation of the question context (i.e. the question and candidate answer), and then uses a reasoning process which combines that vector with a knowledge graph. The resulting reasoning process uses a Graph Neural Network to dynamically update the question context vector based on the symbolic

knowledge captured by the KG. DeepProbLog [42] is also aimed at reasoning about the outputs of a neural network model, in this case based on symbolic probabilistic rules. The general idea of adding a differentiable reasoner on top of a deep neural network model has been explored from a number of other angles. For instance, [67] relies on a differentiable SAT solver to enable reasoning with neural network outputs, while [68] proposes a strategy for using combinatorial optimisation algorithms within an end-to-end differentiable model.

4.2 Complex Query Answering

Learning knowledge graph embeddings has proven a successful approach to predict missing or unobserved edges in knowledge graphs. However, while dealing with knowledge graphs, one is usually interested in handling complex queries describing complex information in the form of graph patterns rather than simple atomic edge-like queries. Indeed, one of the main benefits of symbolically encoded knowledge graphs is that they support SPARQL or conjunctive queries (CQs) [12, 13]. However, symbolically encoded KGs can only be queried for existing facts in the knowledge graph, that is, missing entities or edges cannot be inferred. To address this shortcoming, recently various investigations on the use of knowledge graph embeddings to make predictions about conjunctive queries and extensions thereof on incomplete knowledge graphs have been carried out [69–75]. In this setting, for instance, given an incomplete university knowledge graph, we might want to predict *which students are (likely) attending Math and CS modules that use linear algebra?* Unlike for edge (link) prediction, the query might involve several unobserved edges and entities, effectively making this a more complex problem as there exist a combinatorial number of possible interesting queries, and each of them could be matched to many (unobserved) subgraphs of the input KG. In fact, it is not hard to see that a naive approach to query prediction might be unfeasible in practice [69]. One could first use an edge prediction model on all possible pairs of nodes, and then using the obtained edge likelihood, and then enumerate and score all candidate subgraphs that might satisfy a query. However, this enumeration approach is in the worst-case exponential in the number of existentially quantified variables in the query. As a solution, these works represent KG entities and queries in a joint embedding space. For example, the seminal graph query embedding model (GQE) [69] represents KG entities x and a query q as vectors and then cosine similarity is used to score the plausibility of x being a possible answer to q. Most existing query embedding approaches work compositionally by building the embedding of a query q based on its sub-queries. For example, if the input query q is of the form $q_1 \wedge q_2$, the embedding of q is computed based on the embeddings of q_1 and q_2. A number of these works have concentrated on developing query embeddings that support extensions of conjunctive queries, such as positive existential queries (extending CQs with disjunction) [70] or even existential queries with negation [71]. Recently, [75] proposed a framework for answering positive existential queries using pre-trained link predictors to score the atomic queries composing the input query, which is then evaluated using continuous versions of

logical operators and combinatorial search or gradient descent. Importantly, this work shows that state-of-the-art results can be obtained using a simple framework requiring only neural link predictors trained for atomic queries, rather than millions of queries as in previous works. In all the works mentioned so far it is assumed that queries have a unique missing entity (answer variable). To overcome this shortcoming, [73] proposed an approach based on transformers to deal with conjunctive queries with multiple missing entities. Finally, [72] investigates whether some of the existing query embedding models are *logically faithful* in the sense that they behave like symbolic logical inference systems with respect to entailed answers. They show that existing models behave poorly in finding logically entailed answers, and propose a model improving faithfulness without losing generalization capacity.

4.3 Using Embeddings for Flexible Symbolic Reasoning

In applications where interpretability is important, using symbolic representations is often preferable. For this reason, developing rule based classifiers remains an important topic of research [76, 77, 35]. One important disadvantage, however, is that rule bases are usually incomplete. Indeed, learned rules typically only cover situations that are witnessed (sufficiently frequently) in the training data. Neural network models, on the other hand, have the ability to *interpolate* between such situations, which intuitively allows them to make meaningful predictions across a wider range of situations. When rules are manually provided by a domain expert, beyond toy domains we can usually not expect the resulting rule base to be exhaustive either. To address this concern, a number of methods have been proposed which combine the inductive generalisation abilities of neural models, to allow some form of flexible rule-based reasoning. A standard solution is to use vector representations to implement a form of similarity based reasoning [78, 79]. Consider, for instance, the following rule: *strawberry* $\rightarrow$ *healthy*, and suppose that our knowledge base says nothing about raspberries. Given a standard word embedding [80], we can find out that *strawberry* and *raspberry* are highly similar. Based on this knowledge, we can infer that *raspberry* $\rightarrow$ *healthy* is plausible. However, it is difficult to relate degrees of similarity to the plausibility of the inferred rules in a principled way. For this reason, *interpolation* has been put forward as an alternative to similarity based reasoning [81–83]. The intuition is to start from a minimum of two rules e.g. *raspberry* $\rightarrow$ *healthy* and *orange* $\rightarrow$ *healthy*. Plausible inferences are then supported by the notion of *conceptual betweenness*: we say that a concept B is between the concepts $A_1, ...A_n$ if properties that hold for all of $A_1, ..., A_n$ are also expected to hold for B. If we know that *raspberry* is between *strawberry* and *orange*, then we can plausibly infer the rule *raspberry* $\rightarrow$ *healthy* from the two given ones. This interpolation principle is closely related to the notion of category based induction from cognitive science [84]. While this is a general principle, which can be instantiated in different ways, good results have been obtained using strategies which infer betweenness relations from word embeddings and related vector representations [82, 85].

5 Concluding Remarks

While it seems clear that future AI systems will somehow need to combine the advantages of symbolic and neural representations, the lack of sufficiently comprehensive symbolic knowledge bases, especially those which capture generic and commonsense knowledge, remains an important obstacle. In the last few years, the focus has somewhat shifted from embedding symbolic knowledge bases to learning knowledge about the world by training deep neural language models. The amount of world knowledge captured by the largest models, such as GPT-3 [86], has been particularly surprising. While by no means perfect, even the commonsense reasoning abilities of these models surpasses expectations[1]. To deal with aspects of commonsense knowledge that are rarely stated in text, a typical strategy in recent years has been to crowdsource targeted natural language assertions and explanations [11, 87], and to use such crowdsourced knowledge for fine-tuning language models. However, despite their impressive abilities, neural language models still have two fundamental limitations, which suggest that symbolic representations and systematic reasoning will still play an important role in future AI systems. First, while current NLP models achieve strong results in tasks such as question answering, it is difficult to differentiate between cases where they "know" the answer and cases where they are essentially guessing. Indeed, recent analysis has suggested that language models are still relying on rather shallow heuristics for answering questions [88], which tend to perform well on most benchmarks but offer little in terms of guarantees. Along similar lines, neural machine translation systems are prone to "hallucinating" [89], i.e. generating fluent sentences in the target language which are disconnected from the source text. To use of neural models to make critical decisions thus remains problematic. A second limitation of neural models concerns situations where some form of systematic reasoning is needed. While neural language models can be trained to simulate forward chaining in synthetic settings [59], in practice considerable care is needed to extract the most relevant premises and presenting these in a suitable way, a problem which is studied under the umbrella of multi-hop question answering [90]. Moreover, further progress will need NLP systems to carry out forms of reasoning that go beyond forward or backward chaining, including reasoning about disjunctive knowledge (e.g. arising from the ambiguity of language) and reasoning about the beliefs and intentions of the different participants of a story. It seems unlikely that neural models will be able to carry out such forms for reasoning without relying on some kind of systematic process and structured representation. In fact, for answering questions which require commonsense reasoning, some authors have already found that language models can be improved by repeatedly querying them in a systematic way to extract relevant background knowledge, before trying to answer the question [91, 34].

[1] https://cs.nyu.edu/ davise/papers/GPT3CompleteTests.html

References

1. Lake, B.M., Baroni, M.: Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In: Proceedings of the 35th International Conference on Machine Learning. (2018) 2879–2888
2. Geiger, A., Cases, I., Karttunen, L., Potts, C.: Posing fair generalization tasks for natural language inference. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. (2019) 4484–4494
3. Sinha, K., Sodhani, S., Dong, J., Pineau, J., Hamilton, W.L.: CLUTRR: A diagnostic benchmark for inductive reasoning from text. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. (2019) 4505–4514
4. d'Avila Garcez, A.S., Broda, K., Gabbay, D.M.: Neural-Symbolic Learning Systems – Foundations and Applications. Perspectives in Neural Computing. Springer (2002)
5. d'Avila Garcez, A.S., Gori, M., Lamb, L.C., Serafini, L., Spranger, M., Tran, S.N.: Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. FLAP **6**(4) (2019) 611–632
6. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10) (2014) 78–85
7. Rebele, T., Suchanek, F., Hoffart, J., Biega, J., Kuzey, E., Weikum, G.: YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In: International Semantic Web Conference. (2016) 177–185
8. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: DBpedia – A crystallization point for the web of data. Journal of Web Semantics **7**(3) (2009) 154–165
9. IV, R.L.L., Liu, N.F., Peters, M.E., Gardner, M., Singh, S.: Barack's wife Hillary: Using knowledge graphs for fact-aware language modeling. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. (2019) 5962–5971
10. Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: An open multilingual graph of general knowledge. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2017)
11. Sap, M., Le Bras, R., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N.A., Choi, Y.: ATOMIC: An atlas of machine commonsense for if-then reasoning. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2019) 3027–3035
12. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutiérrez, C., Gayo, J.E.L., Kirrane, S., Neumaier, S., Polleres, A., Navigli, R., Ngomo, A.N., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J.F., Staab, S., Zimmermann, A.: Knowledge graphs. CoRR **abs/2003.02320** (2020)
13. Hogan, A.: Knowledge graphs: Research directions. In: Reasoning Web. Volume 12258 of Lecture Notes in Computer Science., Springer (2020) 223–253
14. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. (2019) 4171–4186
15. Trouillon, T., Dance, C.R., Gaussier, É., Welbl, J., Riedel, S., Bouchard, G.: Knowledge graph completion via complex tensor factorization. J. Mach. Learn. Res. **18** (2017) 130:1–130:38

16. Sun, Z., Deng, Z., Nie, J., Tang, J.: RotatE: Knowledge graph embedding by relational rotation in complex space. In: 7th International Conference on Learning Representations. (2019)
17. Garg, D., Ikbal, S., Srivastava, S.K., Vishwakarma, H., Karanam, H., Subramaniam, L.V.: Quantum embedding of knowledge for reasoning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., eds.: Advances in Neural Information Processing Systems. (2019)
18. Balazevic, I., Allen, C., Hospedales, T.: Multi-relational poincaré graph embeddings. Advances in Neural Information Processing Systems **32** (2019) 4463–4473
19. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. In: Advances in Neural Information Processing Systems. (2019) 2731–2741
20. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems. (2013) 2787–2795
21. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the 3rd International Conference on Learning Representations. (2015)
22. Jain, P., Rathi, S., Mausam, Chakrabarti, S.: Knowledge base completion: Baseline strikes back (again). CoRR **abs/2005.00804** (2020)
23. Kazemi, S.M., Poole, D.: SimplE embedding for link prediction in knowledge graphs. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. (2018) 4289–4300
24. Abboud, R., Ceylan, İ.İ., Lukasiewicz, T., Salvatori, T.: BoxE: A box embedding model for knowledge base completion. In: NeurIPS. (2020)
25. Rogers, A., Kovaleva, O., Rumshisky, A.: A primer in bertology: What we know about how BERT works. Trans. Assoc. Comput. Linguistics **8** (2020) 842–866
26. Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.: Language models as knowledge bases? In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). (2019) 2463–2473
27. Roberts, A., Raffel, C., Shazeer, N.: How much knowledge can you pack into the parameters of a language model? In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. (2020) 5418–5426
28. Geva, M., Schuster, R., Berant, J., Levy, O.: Transformer feed-forward layers are key-value memories. arXiv preprint arXiv:2012.14913 (2020)
29. Dai, D., Dong, L., Hao, Y., Sui, Z., Wei, F.: Knowledge neurons in pretrained transformers. arXiv preprint arXiv:2104.08696 (2021)
30. Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., Tang, J.: KEPLER: A unified model for knowledge embedding and pre-trained language representation. Transactions of the Association for Computational Linguistics **9** (2021) 176–194
31. Yasunaga, M., Ren, H., Bosselut, A., Liang, P., Leskovec, J.: QA-GNN: reasoning with language models and knowledge graphs for question answering. CoRR **abs/2104.06378** (2021)
32. Yao, L., Mao, C., Luo, Y.: KG-BERT: BERT for knowledge graph completion. CoRR **abs/1909.03193** (2019)
33. Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., Choi, Y.: COMET: Commonsense transformers for automatic knowledge graph construction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. (2019) 4762–4779

34. Bosselut, A., Bras, R.L., Choi, Y.: Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In: Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence. (2021) 4923–4931
35. Meilicke, C., Chekol, M.W., Ruffinelli, D., Stuckenschmidt, H.: Anytime bottom-up rule learning for knowledge graph completion. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. (2019) 3137–3143
36. Sadeghian, A., Armandpour, M., Ding, P., Wang, D.Z.: DRUM: end-to-end differentiable rule mining on knowledge graphs. In Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R., eds.: Proceedings of the Annual Conference on Neural Information Processing Systems. (2019) 15321–15331
37. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach (4th Edition). Pearson (2020)
38. Rocktäschel, T., Riedel, S.: End-to-end differentiable proving. In: Proceedings of the Annual Conference on Neural Information Processing Systems. (2017) 3788–3800
39. Sourek, G., Aschenbrenner, V., Zelezný, F., Schockaert, S., Kuzelka, O.: Lifted relational neural networks: Efficient learning of latent relational structures. J. Artif. Intell. Res. **62** (2018) 69–100
40. Donadello, I., Serafini, L., d'Avila Garcez, A.S.: Logic tensor networks for semantic image interpretation. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. (2017) 1596–1602
41. Xu, J., Zhang, Z., Friedman, T., Liang, Y., den Broeck, G.V.: A semantic loss function for deep learning with symbolic knowledge. In: Proceedings of the 35th International Conference on Machine Learning. (2018) 5498–5507
42. Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., Raedt, L.D.: DeepProbLog: Neural probabilistic logic programming. In: Proceedings of the Annual Conference on Neural Information Processing Systems. (2018) 3753–3763
43. Yang, Z., Ishay, A., Lee, J.: NeurASP: Embracing neural networks into answer set programming. In: IJCAI, ijcai.org (2020) 1755–1762
44. Dai, W., Xu, Q., Yu, Y., Zhou, Z.: Bridging machine learning and logical reasoning by abductive learning. In: NeurIPS. (2019) 2811–2822
45. Tsamoura, E., Hospedales, T.M., Michael, L.: Neural-symbolic integration: A compositional perspective. In: AAAI, AAAI Press (2021) 5051–5060
46. Evans, R., Grefenstette, E.: Learning explanatory rules from noisy data. Journal of Artificial Intelligence Research **61** (2018) 1–64
47. Yang, F., Yang, Z., Cohen, W.W.: Differentiable learning of logical rules for knowledge base reasoning. In: Proceedings of the Annual Conference on Neural Information Processing Systems 2017. (2017) 2319–2328
48. Gärdenfors, P.: Conceptual Spaces - The Geometry of Thought. MIT Press (2000)
49. Gutiérrez-Basulto, V., Schockaert, S.: From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules. In: Proceedings of the Sixteenth International Conference on Principles of Knowledge Representation and Reasoning. (2018) 379–388
50. Patel, D., Dasgupta, S.S., Boratko, M., Li, X., Vilnis, L., McCallum, A.: Representing joint hierarchies with box embeddings. In: Proceedings of the Conference on Automated Knowledge Base Construction. (2020)
51. Ganea, O., Bécigneul, G., Hofmann, T.: Hyperbolic entailment cones for learning hierarchical embeddings. In: Proceedings of the International Conference on Machine Learning. (2018) 1646–1655

52. Özçep, Ö.L., Leemhuis, M., Wolter, D.: Cone semantics for logics with negation. In: Proceedings of the International Joint Conference on Artificial Intelligence. (2020) 1820–1826
53. Garg, D., Ikbal, S., Srivastava, S.K., Vishwakarma, H., Karanam, H.P., Subramaniam, L.V.: Quantum embedding of knowledge for reasoning. In: Proceedings of the Annual Conference on Neural Information Processing Systems. (2019) 5595–5605
54. Gärdenfors, P., Williams, M.: Reasoning about categories in conceptual spaces. In: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence. (2001) 385–392
55. Goldstone, R.L.: Isolated and interrelated concepts. Memory & Cognition **24**(5) (1996) 608–628
56. Bouraoui, Z., Camacho-Collados, J., Anke, L.E., Schockaert, S.: Modelling semantic categories using conceptual neighborhood. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence. (2020) 7448–7455
57. Talmor, A., Tafjord, O., Clark, P., Goldberg, Y., Berant, J.: Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. In: Proceedings of the Annual Conference on Neural Information Processing Systems. (2020)
58. Kassner, N., Krojer, B., Schütze, H.: Are pretrained language models symbolic reasoners over knowledge? In: Proceedings of the 24th Conference on Computational Natural Language Learning. (2020) 552–564
59. Clark, P., Tafjord, O., Richardson, K.: Transformers as soft reasoners over language. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. (2020) 3882–3890
60. Zha, H., Chen, Z., Yan, X.: Inductive relation prediction by BERT. CoRR **abs/2103.07102** (2021)
61. Rocktäschel, T., Singh, S., Riedel, S.: Injecting logical background knowledge into embeddings for relation extraction. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. (2015) 1119–1129
62. Hu, Z., Ma, X., Liu, Z., Hovy, E.H., Xing, E.P.: Harnessing deep neural networks with logic rules. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. (2016)
63. Li, T., Srikumar, V.: Augmenting neural networks with first-order logic. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. (2019) 292–302
64. Ratner, A., Bach, S.H., Ehrenberg, H.R., Fries, J.A., Wu, S., Ré, C.: Snorkel: rapid training data creation with weak supervision. VLDB J. **29**(2-3) (2020) 709–730
65. Awasthi, A., Ghosh, S., Goyal, R., Sarawagi, S.: Learning from rules generalizing labeled exemplars. In: Proceedings of the 8th International Conference on Learning Representations. (2020)
66. Yasunaga, M., Ren, H., Bosselut, A., Liang, P., Leskovec, J.: QA-GNN: reasoning with language models and knowledge graphs for question answering. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. (2021) 535–546
67. Wang, P.W., Donti, P., Wilder, B., Kolter, Z.: SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In: Proceedings of the International Conference on Machine Learning. (2019) 6545–6554
68. Niepert, M., Minervini, P., Franceschi, L.: Implicit MLE: backpropagating through discrete exponential family distributions. CoRR **abs/2106.01798** (2021)

69. Hamilton, W.L., Bajaj, P., Zitnik, M., Jurafsky, D., Leskovec, J.: Embedding logical queries on knowledge graphs. In: NeurIPS. (2018) 2030–2041
70. Ren, H., Hu, W., Leskovec, J.: Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In: ICLR, OpenReview.net (2020)
71. Ren, H., Leskovec, J.: Beta embeddings for multi-hop logical reasoning in knowledge graphs. In: NeurIPS. (2020)
72. Sun, H., Arnold, A.O., Bedrax-Weiss, T., Pereira, F., Cohen, W.W.: Faithful embeddings for knowledge base queries. In: NeurIPS. (2020)
73. Kotnis, B., Lawrence, C., Niepert, M.: Answering complex queries in knowledge graphs with bidirectional sequence encoders. In: AAAI, AAAI Press (2021) 4968–4977
74. Choudhary, N., Rao, N., Katariya, S., Subbian, K., Reddy, C.K.: Self-supervised hyperboloid representations from logical queries over knowledge graphs. In: WWW, ACM / IW3C2 (2021) 1373–1384
75. Arakelyan, E., Daza, D., Minervini, P., Cochez, M.: Complex query answering with neural link predictors. In: Proceedings of the 9th International Conference on Learning Representations. (2021)
76. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.: AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of the 22nd International Conference on World Wide Web. (2013) 413–422
77. Omran, P.G., Wang, K., Wang, Z.: Scalable rule learning via learning representation. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. (2018) 2149–2155
78. d'Amato, C., Fanizzi, N., Fazzinga, B., Gottlob, G., Lukasiewicz, T.: Ontology-based semantic search on the web and its combination with the power of inductive reasoning. Ann. Math. Artif. Intell. **65**(2-3) (2012) 83–121
79. Beltagy, I., Chau, C., Boleda, G., Garrette, D., Erk, K., Mooney, R.J.: Montague meets Markov: Deep semantics with probabilistic logical form. In: Proceedings of the Second Joint Conference on Lexical and Computational Semantics. (2013) 11–21
80. Mikolov, T., Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings NAACL-HLT. (2013) 746–751
81. Schockaert, S., Prade, H.: Interpolative and extrapolative reasoning in propositional theories using qualitative knowledge about conceptual spaces. Artif. Intell. **202** (2013) 86–131
82. Derrac, J., Schockaert, S.: Inducing semantic relations from conceptual spaces: A data-driven approach to plausible reasoning. Artif. Intell. **228** (2015) 66–94
83. Ibáñez-García, Y., Gutiérrez-Basulto, V., Schockaert, S.: Plausible reasoning about el-ontologies using concept interpolation. In: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning. (2020) 506–516
84. Osherson, D.N., Smith, E.E., Wilkie, O., Lopez, A., Shafir, E.: Category-based induction. Psychological Review **97**(2) (1990) 185–200
85. Bouraoui, Z., Schockaert, S.: Automated rule base completion as bayesian concept induction. In: The Thirty-Third AAAI Conference on Artificial Intelligence. (2019) 6228–6235
86. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark,

J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., eds.: Proceedings of the Annual Conference on Neural Information Processing Systems. (2020)
87. Mostafazadeh, N., Kalyanpur, A., Moon, L., Buchanan, D.W., Berkowitz, L., Biran, O., Chu-Carroll, J.: GLUCOSE: generalized and contextualized story explanations. In Webber, B., Cohn, T., He, Y., Liu, Y., eds.: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. (2020) 4569–4586
88. Dufter, P., Kassner, N., Schütze, H.: Static embeddings as efficient knowledge bases? In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y., eds.: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. (2021) 2353–2363
89. Raunak, V., Menezes, A., Junczys-Dowmunt, M.: The curious case of hallucinations in neural machine translation. In Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y., eds.: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. (2021) 1172–1183
90. Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., Manning, C.D.: Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2018) 2369–2380
91. Shwartz, V., West, P., Bras, R.L., Bhagavatula, C., Choi, Y.: Unsupervised commonsense question answering with self-talk. In Webber, B., Cohn, T., He, Y., Liu, Y., eds.: Proceedings of the Conference on Empirical Methods in Natural Language Processing. (2020) 4615–4629