Integrating Ontologies and Vector Space Embeddings using Conceptual Spaces

Zied Bouraoui ✉
CRIL Laboratory, Université d'Artois, France

Víctor Gutiérrez-Basulto ✉
School of Computer Science & Informatics, Cardiff University, UK

Steven Schockaert ✉
School of Computer Science & Informatics, Cardiff University, UK

Abstract

Ontologies and vector space embeddings are among the most popular frameworks for encoding conceptual knowledge. Ontologies excel at capturing the logical dependencies between concepts in a precise and clearly defined way. Vector space embeddings excel at modelling similarity and analogy. Given these complementary strengths, there is a clear need for frameworks that can combine the best of both worlds. In this paper, we present an overview of our recent work in this area. We first discuss the theory of conceptual spaces, which was proposed in the 1990s by Gärdenfors as an intermediate representation layer in between embeddings and symbolic knowledge bases. We particularly focus on a number of recent strategies for learning conceptual space representations from data. Next, building on the idea of conceptual spaces, we discuss approaches where relational knowledge is modelled in terms of geometric constraints. Such approaches aim at a tight integration of symbolic and geometric representations, which unfortunately comes with a number of limitations. For this reason, we finally also discuss methods in which similarity, and other forms of conceptual relatedness, are derived from vector space embeddings and subsequently used to support flexible forms of reasoning with ontologies, thus enabling a looser integration between embeddings and symbolic knowledge.

2012 ACM Subject Classification Computing methodologies → Knowledge representation and reasoning

Keywords and phrases Conceptual Spaces, Ontologies, Vector Space Embeddings, Learning and Reasoning

Digital Object Identifier 10.4230/OASIcs.AIB.2022.3

Funding This work has been supported by a grant from the Leverhulme Trust (project RPG-2021-140).

1 Introduction

In Artificial Intelligence (AI), the traditional approach for encoding knowledge about concepts has been to use logic-based representations, typically in the form of a rule base. Such a rule base is often called an ontology in this context.

▶ **Example 1.** Consider the following rules:

$$\mathsf{expertInAI}(X) \leftarrow \mathsf{authorOf}(X, Y), \mathsf{hasTopic}(Y, \mathsf{artificialIntelligence})$$

$$\mathsf{hasTopic}(X, \mathsf{artificialIntelligence}) \leftarrow \mathsf{hasTopic}(X, \mathsf{knowledgeRepresentation})$$

$$\mathsf{hasTopic}(X, \mathsf{artificialIntelligence}) \leftarrow \mathsf{hasTopic}(X, \mathsf{machineLearning})$$

$$\mathsf{hasTopic}(X, \mathsf{artificialIntelligence}) \leftarrow \mathsf{hasTopic}(X, \mathsf{multiAgentSystems})$$

$$\mathsf{hasTopic}(X, \mathsf{artificialIntelligence}) \leftarrow \mathsf{hasTopic}(X, \mathsf{naturalLanguageProcessing})$$

International Research School in Artificial Intelligence in Bergen (AIB 2022).
Editors: Camille Bourgaux, Ana Ozaki, and Rafael Peñaloza; Article No. 3; pp. 3:1–3:30
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Here we have used the notational conventions from logic programming, where the conclusion of the rule is shown on the left-hand side and "," denotes conjunction. The first rule intuitively asserts that somebody who has published a paper on an AI topic is an expert in AI. The remaining rules encode that knowledge representation, machine learning, multi-agent systems and natural language processing are sub-fields of AI. Along with the ontology, we are usually given a set of facts, e.g.:

$$\{\mathsf{authorOf}(\mathsf{bob}, p), \mathsf{hasTopic}(p, \mathsf{knowledgeRepresentation})\}$$

Given this set of facts, together with the aforementioned rules, we can conclude that $\mathsf{hasTopic}(p, \mathsf{artificialIntelligence})$ holds and thus also that $\mathsf{expertInAI}(\mathsf{bob})$ holds.

Using ontologies for encoding conceptual knowledge has at least two key advantages. First, the formal semantics of the underlying logic ensures that knowledge can be encoded in a precise and unambiguous way. This, in turn, ensures that different applications can rely on a shared understanding of the meaning of the concepts involved. Second, ontologies enable us to capture knowledge in a transparent and interpretable way[1], which makes it relatively straightforward to update knowledge and to support decisions with meaningful explanations. But ontologies, and symbolic approaches to knowledge representation more generally, also have important drawbacks. A first issue stems from the fact that the knowledge which is captured in an ontology is rarely complete. For instance, consider the following set of facts:

$$\{\mathsf{authorOf}(\mathsf{alice}, q), \mathsf{hasTopic}(q, \mathsf{planning})\}$$

As none of the available rules express that planning is a sub-field of AI, we cannot infer that $\mathsf{expertInAI}(\mathsf{alice})$ holds. Nonetheless, to a human observer, it seems clear that this would be a valid inference, even without a precise understanding of what the predicate $\mathsf{expertInAI}$ is supposed to capture. Essentially, standard frameworks for modelling ontologies lack a mechanism for inductive reasoning [28]. This is not something which can be easily addressed, as inductive arguments rely on graded notions such as similarity and typicality [58, 50, 66, 51]. Another issue is that many concepts are difficult to characterise in a satisfactory way using logical rules. For instance, somebody with a single published paper in AI would not normally be considered to be an AI expert, except perhaps if the work was particularly influential or groundbreaking, but formalising such notions using rules is challenging. Probabilistic extensions of standard ontology languages [36, 15] may alleviate some of the aforementioned issues, but such frameworks still do not allow us to model similarity, or aspects that are a matter of degree (e.g. being an expert in AI).

The most common alternative to ontologies is to encode conceptual knowledge using vector space representations. Most work on vector representations of conceptual knowledge has focused on knowledge graphs (KGs), which are sets of triples of the form (e, r, f), where e and f are entities and r is a binary relation. Note that both individuals and attribute values are typically regarded as entities in this context. As an example, we may consider the following knowledge graph:

$$\begin{aligned} K = \{&(\mathsf{bob}, \mathsf{authorOf}, \mathsf{p}), (p, \mathsf{hasTopic}, \mathsf{knowledgeRepresentation}), \\ &(p, \mathsf{hasTopic}, \mathsf{artificialIntelligence}), (\mathsf{bob}, \mathsf{hasProperty}, \mathsf{expertInAI})\} \end{aligned}$$

[1] It should be noted, however, that the extent to which a given ontology is interpretable will depend on its size and the way it has been encoded. Symbolic rules that have been learned from data can often be difficult to interpret, for instance.

Approaches for Knowledge graph embedding (KGE) learn a vector representation $\mathbf{e} \in \mathbb{R}^n$ for each entity e and a scoring function $\phi_r : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ for each relation type r, such that $\phi_r(\mathbf{e}, \mathbf{f})$ captures the plausibility of the triple (e, r, f), i.e. the plausibility that the relation r holds between the entities e and f [14, 75, 70, 69]. The vector $\mathbf{e}$ is called the *embedding* of entity e. The purpose of KGE is at least two-fold. First, it is hoped that this embedding will uncover some of the underlying semantic dependencies in the KG, and that as a result, we will be able to identify plausible triples that are missing from the given KG. Second, by encoding the information that is captured in the knowledge graph using vectors, it becomes easier to exploit this information in neural network models.

Figure 1 shows a vector encoding of the paper p and some of the considered subject areas. For this example, we assume that the dot product between p and a subject area indicates how relevant that subject area is to p, i.e. we have $\phi_{\mathsf{hasTopic}}(\mathbf{e}, \mathbf{f}) = \mathbf{e} \cdot \mathbf{f}$. Let us write $\mathbf{v_{ML}}$, $\mathbf{v_{AI}}$, $\mathbf{v_{NLP}}$ and $\mathbf{v_{KR}}$ for the vector representations of the different subject areas, and $\mathbf{p}$ for the representation of p. According to this embedding, we have $\mathbf{p} \cdot \mathbf{v_{ML}} \approx \mathbf{p} \cdot \mathbf{v_{NLP}} > \mathbf{p} \cdot \mathbf{v_{KR}}$, which captures the knowledge that p is more closely related to machine learning and natural language processing than to knowledge representation. Moreover, note how the norm of $\mathbf{v_{AI}}$ is larger than the norms of $\mathbf{v_{ML}}$, $\mathbf{v_{NLP}}$ and $\mathbf{v_{KR}}$. This intuitively captures the knowledge that the term artificial intelligence is broader in meaning. For instance, we can encode the knowledge that machine learning is a sub-discipline of AI by ensuring that for every vector $\mathbf{x} \in \mathbb{R}^2$ it holds that:

$$\mathbf{v_{ML}} \cdot \mathbf{x} < \mathbf{v_{AI}} \cdot \mathbf{x}$$

Note that in this example, we have only focused on one relation (i.e. hasTopic). In general, we can model multiple relations by using higher-dimensional vectors, together with scoring functions that depend on relation-specific parameters (see Section 2.3 for more details). When it comes to modelling conceptual knowledge, an important advantage of KGE is that it naturally supports inductive inferences. Moreover, such representations are better suited for modelling graded notions such as similarity than symbolic representations. However, the extent to which "rule-like" knowledge can be captured is limited. As we saw in the aforementioned example, we can model the fact that one concept is subsumed by another, but it is not clear how more complex rules can be encoded using vector space embeddings. Moreover, KGE models lack the transparency of symbolic representations, which makes it harder to generate meaningful explanations or to update representations (e.g. to correct mistakes, add new knowledge, or take account of changes in the world).

It is thus clear that ontologies and vector space embeddings have complementary strengths and weaknesses when it comes to modelling conceptual knowledge. Accordingly, various authors have proposed strategies for combining these two paradigms. For instance, rules are sometimes used to regularise neural networks [24, 74, 43], to generate supplementary training data [7], or to determine the structure of a neural network [59, 67]. Other approaches use rules to reason about the predictions of neural networks [44, 77], or treat rules as latent variables which are inferred by a neural network [56]. Note, however, how in the aforementioned research lines, rules and vector representation are treated as fundamentally distinct. Rules are either used as a supervision signal for learning neural networks (or vector space embeddings) or they are used for reasoning in a way that is largely decoupled from the neural networks or vector space embeddings themselves. Another observation is that rules essentially play a supportive role, to help overcome the limitations of some neural network model.

The first question we address in this paper is whether a tighter integration of rules and vector representations is possible. The main idea is to view symbolic knowledge as qualitative constraints on some underlying geometric model. This idea was developed in the 1990s by

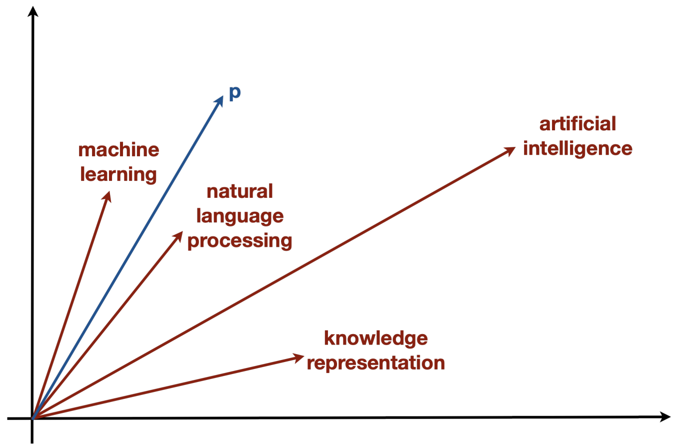


Figure 1 Illustration of a simple knowledge graph embedding, in which the dot product between p and a subject area indicates how relevant that subject area is to p.

Gärdenfors in his theory of conceptual spaces [27]. The key characteristic of conceptual spaces is that concepts are represented as regions, rather than vectors. A rule $A(x) \leftarrow B(x), C(x)$ can then be viewed as the constraint that the intersection of the regions representing B and C should be included in the region representing A. While the theory of conceptual spaces offers an elegant solution to the question of how symbolic and vector representation could be integrated, it has two limitations that have hampered its adoption within AI:

- In practice, it is often difficult to learn region-based representations of concepts from data.
- Conceptual space representations cannot be used for modelling relational knowledge, e.g. rules involving binary predicates.

These two limitations, and strategies for addressing them, are discussed in Sections 3 and 4.

The second question we discuss is how vector space representations can be used in a supportive role, to help overcome some of the limitations of symbolic reasoning with ontologies. Here, the starting point is that some of the aforementioned shortcomings can be alleviated within a purely symbolic setting, for instance by relying on default reasoning [42, 20, 32], analogical reasoning [31, 54, 61], or qualitative versions of similarity based reasoning [65, 63]. The main problem with implementing such strategies in practice comes from the fact that they often rely on types of background knowledge which is not usually available in symbolic form (e.g. qualitative similarity relations). However, in some cases, this background knowledge can be obtained from vector space embeddings. In this case, we still have a loose integration between vector representations and rules, but rather than trying to improve neural network learning, as in the works described above, now the focus is on making symbolic reasoning more flexible and adding some kind of inductive reasoning capability. For instance, in the setting from Example 1, if we know that the vector representation of planning is highly similar to the vector representation of knowledgeRepresentation, we can plausibly infer the following rule:

$$\mathsf{hasTopic}(X, \mathsf{artificialIntelligence}) \leftarrow \mathsf{hasTopic}(X, \mathsf{planning})$$

In Section 5, we discuss a number of strategies that build on this idea, focusing on how such plausible inferences can be integrated with standard deductive reasoning.

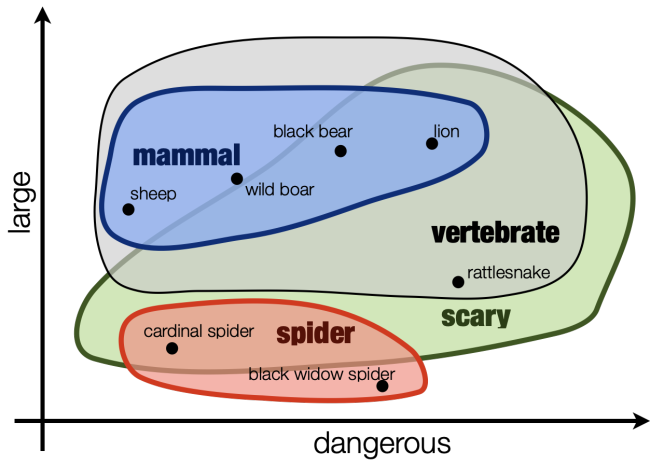


Figure 2 Illustration of a conceptual space of animals.

2 Background

In this section, we briefly introduce the main concepts that we will build on in the following sections. First, Section 2.1 discusses the theory of conceptual spaces. In Section 2.2 we then cover two standard formalisms for encoding ontological rules: existential rules and the $\mathcal{EL}$-family of description logics. Finally, Section 2.3 provides an introduction into Knowledge Graph Embedding.

2.1 Conceptual Spaces

Similar to vector-space embeddings, conceptual spaces [27] are geometric representations of the entities from a given domain of discourse. However, conceptual spaces differ from standard embeddings in two important ways: (i) properties and concepts are represented as regions and (ii) the dimensions of a conceptual space correspond to semantically meaningful features. These two differences enable conceptual spaces to act as an interface between neural representations, on the one hand, and symbolic knowledge, on the other hand. This is illustrated in Figure 2, which shows a conceptual space of animals. Specific animals are represented as points in this space. Concepts such as mammal and properties such as scary are represented as regions. The dimensions capture the ordinal features dangerous and large. In this representation, the region modelling mammal is included in the region modelling vertebrate, which intuitively captures the rule $\mathsf{vertebrate}(X) \leftarrow \mathsf{mammal}(X)$, i.e. all mammals are vertebrates. Note how this representation can also capture semantic dependencies that are harder to encode using rules, e.g. the fact that large spiders are scary.

While it is convenient to think about conceptual spaces as vector space embeddings with some added structure, conceptual spaces do not necessarily have the structure of a vector space. A conceptual space is defined from a set of *quality dimensions* $Q_1, ..., Q_n$. Each of these quality dimensions captures a primitive feature. As a standard example, the conceptual space of colours is built from three quality dimensions, representing hue, saturation and intensity. A distinction is made between *integral* and *separable* quality dimensions. Intuitively, separable quality dimensions are those that have a meaning on their own. For instance, *size* could be seen as a separable dimension. On the other hand, *hue* is not separable, as we cannot imagine the hue of a colour without also specifying its saturation and intensity. This distinction between integral and separable dimensions plays an important role in cognitive theories, as it affects how similarity is perceived. For instance, Euclidean distance is normally used when integral dimensions need to be combined, whereas Manhattan distance is used when separable dimensions need to be combined [49, 27]. Quality dimensions are partitioned into so-called *domains*, where dimensions that belong to the same domain are assumed to be integral, while dimensions from different domains are assumed to be separable. For instance, a conceptual space of physical objects may be composed of three domains: the colour domain (containing the hue, saturation and intensity quality dimensions), the size domain (containing only a single quality dimension) and the shape domain (containing several dimensions).

We can view domains as Cartesian products of quality dimensions. For instance, if D_i is composed of the quality dimensions $Q_1, ..., Q_k$ then the elements of D_i are tuples $(x_1, ..., x_k) \in Q_1 \times ... \times Q_k$. We can thus intuitively think of domains as vector spaces, although in general it is not required that domains satisfy the axioms of a vector space. An individual (e.g. a specific apple) is represented as an element $(x_1, ..., x_k)$ of a given domain, whereas we can think of properties (e.g. green) as regions. One of the central assumptions in the theory of conceptual spaces is that each *natural* property corresponds to a *convex* region in some domain. A *concept* is characterised in terms of a set of natural properties, along with information about how these properties are correlated. To define this notion of convexity, we have to assume that each domain D_i is equipped with a ternary betweenness relation $\mathsf{bet}_i \subseteq D_i \times D_i \times D_i$. A region $R \subseteq D_i$ is then said to be *convex* iff

$$\forall a, b, c \in D_i \,.\, a \in D_i \wedge c \in D_i \wedge \mathsf{bet}_i(a, b, c) \Rightarrow b \in D_i$$

In this paper, our focus will be on learning conceptual spaces from data. In this case, we will only consider domains that correspond to Euclidean spaces, where the notion of convexity can be interpreted in the standard way. Our focus will be on (i) learning region based representations of properties and concepts (ii) identifying quality-dimensions and (iii) grouping these quality-dimensions into domains.

2.2 Ontology Languages

We next look at two of the most popular Horn-like formalisms to encode ontologies, namely existential rules [10, 35] and the $\mathcal{EL}$-family of description logics [8]. Informally, an existential rule is a datalog-like rule (i.e. a logic programming rule of the kind we used in Example 1) with existentially quantified variables in the head, i.e. it extends traditional datalog with *value invention.* As a consequence, existential rules describe not only constraints on the currently available knowledge or data, but also *intensional* knowledge about the domain of discourse. Likewise, the $\mathcal{EL}$-family of description logics can be used for modelling intentional knowledge. In fact, some expressive members of the $\mathcal{EL}$-family are restrictions of existential rules to unary and binary relations.

Existential Rules

Syntax Let $\mathbf{C}, \mathbf{N}$ and $\mathbf{V}$ be infinite disjoint sets of *constants, (labelled) nulls* and *variables*, respectively. A *term* t is an element in $\mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$; an *atom* α is an expression of the form $R(t_1, \ldots, t_n)$, where R is a *relation name (or predicate)* with *arity* n and terms t_i. An *existential rule* σ is an expression of the form

$$\exists X_1, \ldots, X_j . H_1 \wedge \ldots \wedge H_k \leftarrow B_1 \wedge \ldots \wedge B_n, \quad (1)$$

where $n \geq 0$, $k \geq 1$, $B_1, \ldots B_n$ and $H_1, \ldots, H_k$ are atoms with terms in $\mathbf{C} \cup \mathbf{V}$, and $X_1, ..., X_j \in \mathbf{V}$. From here on, we assume w.l.o.g. that $k = 1$ [21] and we omit the subscript in H_1. We further allow *negative constraints* (also simply called *constraints*), which are expressions of the form $\bot \leftarrow B_1 \wedge \ldots \wedge B_n$, where the B_is are as above and $\bot$ denotes the truth constant *false*. A finite set Σ of existential rules and constraints is called an *ontology*. Let $\mathfrak{R}$ be a set of relation names. A *database* D is a finite set of *facts* over $\mathfrak{R}$, i.e. atoms with terms in $\mathbf{C}$. A *knowledge base (KB)* $\mathcal{K}$ is a pair (Σ, D) with Σ an ontology and D a database.

Semantics An *interpretation* $\mathcal{I}$ *over* $\mathfrak{R}$ is a (possibly infinite) set of atoms over $\mathfrak{R}$ with terms in $\mathbf{C} \cup \mathbf{N}$. An interpretation $\mathcal{I}$ is a *model of* Σ if it satisfies all rules and constraints: $\{B_1, \ldots, B_n\} \subseteq \mathcal{I}$ implies $\{H\} \subseteq \mathcal{I}$ for every existential rule σ in Σ, where existential variables can be witnessed by constants or labelled nulls, and $\{B_1, \ldots, B_n\} \not\subseteq \mathcal{I}$ for all constraints defined as above in Σ; it is a *model of a database* D if $D \subseteq \mathcal{I}$; it is a model of a KB $\mathcal{K} = (\Sigma, D)$, written $\mathcal{I} \models \mathcal{K}$, if it is a model of Σ and D. We say that a KB $\mathcal{K}$ is satisfiable if it has a model. We refer to elements in $\mathbf{C} \cup \mathbf{N}$ simply as *objects*, call atoms α containing only objects as terms *ground*, and denote with $\mathfrak{O}(\mathcal{I})$ the set of all objects occurring in $\mathcal{I}$.

▶ **Example 2.** Let $D = \{\mathsf{wife}(\mathsf{anna}), \mathsf{wife}(\mathsf{marie})\}$ be a database and Σ an ontology composed by the following existential rules:

$$\mathsf{husband}(Y) \leftarrow \mathsf{wife}(X) \wedge \mathsf{married}(X, Y) \tag{2}$$

$$\exists X . \mathsf{husband}(X) \wedge \mathsf{married}(X, Y) \leftarrow \mathsf{wife}(Y) \tag{3}$$

$$\bot \leftarrow \mathsf{husband}(X) \wedge \mathsf{wife}(X) \tag{4}$$

Then, an example of a model of $\mathcal{K} = (\Sigma, D)$ is the set of atoms

$$D \cup \{\mathsf{husband}(o_1), \mathsf{husband}(o_2), \mathsf{married}(o_1, \mathsf{anna}), \mathsf{married}(o_2, \mathsf{marie})\}$$

where o_i are labelled nulls. Note that e.g. $\{\mathsf{married}(\mathsf{anna}, \mathsf{marie}), \mathsf{husband}(\mathsf{marie})\}$ is not included in any model of $\mathcal{K}$ due to (4).

$\mathcal{EL}$-family

We introduce some basic notions about description logics, focusing on $\mathcal{EL}_\bot$, one of the most commonly used logics from the $\mathcal{EL}$-family. The interested reader can find more details on description logics in [9].

Syntax Consider countably infinite but disjoint sets of *concept names* $\mathsf{N_C}$ and *role names* $\mathsf{N_R}$. These concept and role names are combined to $\mathcal{EL}_\bot$ *concepts*, in accordance with the following grammar, where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$:

$$C, D := \top \mid \bot \mid A \mid C \sqcap D \mid \exists r.C$$

For instance, $A \sqcap (\exists r.(B \sqcap C))$ is an example of a well-formed $\mathcal{EL}_\bot$ concept, assuming $A, B, C \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. The fragment of $\mathcal{EL}_\bot$ in which $\bot$ is not used is known as $\mathcal{EL}$. An $\mathcal{EL}_\bot$ *TBox (ontology)* $\mathcal{T}$ is a finite set of *concept inclusions (CIs)* of the form $C \sqsubseteq D$, where C, D are $\mathcal{EL}_\bot$ concepts.

▶ **Example 3.** The ontology in Example 2 can be expressed using the following $\mathcal{EL}$ concept inclusions

$$\exists \mathsf{married}.\mathsf{Wife} \sqsubseteq \mathsf{Husband} \tag{5}$$

$$\mathsf{Wife} \sqsubseteq \exists \mathsf{married}.\mathsf{Husband} \tag{6}$$

$$\mathsf{Husband} \sqcap \mathsf{Wife} \sqsubseteq \bot \tag{7}$$

Semantics The semantics of description logics are usually given in terms of first-order interpretations $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. Such interpretations consist of a nonempty *domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$, which maps each concept name A to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each role name r to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function $\cdot^{\mathcal{I}}$ is extended to complex concepts as follows:

$$(\top)^{\mathcal{I}} = \Delta^{\mathcal{I}}, \qquad (\bot^{\mathcal{I}}) = \emptyset \qquad (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}},$$

$$(\exists r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \exists d' \in C^{\mathcal{I}}, (d, d') \in r^{\mathcal{I}}\}.$$

We now introduce two classical reasoning tasks. An interpretation $\mathcal{I}$ *satisfies* a concept inclusion $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; it is a model of a concept C if $C^{\mathcal{I}} \neq \emptyset$; it is a *model* of a TBox $\mathcal{T}$ if it satisfies all CIs in $\mathcal{T}$. A concept C *subsumes a concept* D *relative to a TBox* $\mathcal{T}$ if every model $\mathcal{I}$ of $\mathcal{T}$ satisfies $C \sqsubseteq D$. We denote this by writing $\mathcal{T} \models C \sqsubseteq D$. A concept C is *satisfiable w.r.t.* $\mathcal{T}$ if there is a common model of C and $\mathcal{T}$.

2.3 Knowledge Graph Embedding

Let a set of entities $\mathcal{E}$ and a set of binary relations $\mathcal{R}$ be given. A knowledge graph (KG) is a subset of $\mathcal{E} \times \mathcal{R} \times \mathcal{E}$. In other words, a knowledge graph is a set of triples of the form (e, r, f). These triples encode the fact that the relation r holds between the entities e and f. For instance, we may have a triple such as $(\mathsf{london}, \mathsf{capitalOf}, \mathsf{uk})$, encoding that London is the capital of the UK. A knowledge graph is thus essentially a set of relational facts, with the limitation that all relations are binary. Note, however, that the set of entities $\mathcal{E}$ typically includes both individuals (i.e. constants referring to specific objects, e.g. london) and attribute values, which allow us to encode unary predicates. For instance, the relational fact $\mathsf{scary}(\mathsf{lion})$ Could be encoded as the KG triple $(\mathsf{lion}, \mathsf{hasAttribute}, \mathsf{scary})$.

The aim of Knowledge Graph Embedding (KGE) is to learn a vector encoding $\mathbf{e} \in \mathbb{R}^n$ for each $e \in \mathcal{E}$ and a scoring function $\phi_r : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ for each $\in \mathcal{R}$. The vector $\mathbf{e}$ is usually referred to as the *embedding* of e. The scoring function is designed such that $\phi_r(\mathbf{e}, \mathbf{f})$ indicates how likely it is that (e, r, f) is a valid triple, i.e. that the relational fact $r(e, f)$ is true. We may assume, for instance, that for each $r \in \mathcal{R}$ we also have a threshold λ_r such that (e, r, f) is considered to be valid iff $\phi_r(\mathbf{e}, \mathbf{f}) \geq \lambda_r$. A comprehensive overview of knowledge graph embedding models is beyond the scope of this paper; please refer to [72, 60] for more complete introductions. To illustrate the main concepts, we discuss a number of popular models. TransE [14] was one of the first KGE models. Relations in this model are viewed as translations. In particular, each relation $r \in \mathcal{R}$ is represented by a vector $\mathbf{r} \in \mathbb{R}^n$. The corresponding scoring function ϕ_r is given by:

$$\phi_r(e, f) = -d(\mathbf{e} + \mathbf{r}, \mathbf{f})$$

with d either Euclidean or Manhattan distance. Another popular choice is to use a bilinear scoring function. In this case, r is parametrised by a matrix $\mathbf{M_r}$ and we have:

$$\phi_r(e, f) = \mathbf{e}^T \, \mathbf{M_r} \, \mathbf{f}$$

Different models differ in which constraints they put on the matrix $\mathbf{M_r}$. For instance, in the RESCAL model [47] this matrix is unconstrained, whereas DistMult [76] only allows diagonal matrices. In recent years, several authors have focused on designing models that make it easier to capture certain relational structures. For instance, embeddings based on hyperbolic geometry have been used to make it easier to model hierarchical structures, such as *is-a* and *part-of* hierarchies [48]. Region-based models, e.g. representing entities as boxes or cones, have been used for their ability to model both hierarchies and intersections [1, 52, 79]. In [68] a model is proposed in which relations are viewed as rotations, to facilitate modelling relational composition, as well as properties such as symmetry. It should be noted, however, that while these models can capture certain relational dependencies to some extent, in most models there is no explicit link between a given knowledge graph embedding and the relational dependencies it captures. Moreover, relatively little is known about which kinds of dependencies different models are capable of capturing (or, more generally, which sets of dependencies can be jointly captured). Of course, this first requires us to formalise what it means for an embedding to capture a relational dependency. We will return to this question in Section 4.

3 Learning Conceptual Space Representations

If we want to use conceptual spaces as an interface between symbolic ontologies and vectors space embeddings, a crucial question is whether it is possible to learn conceptual spaces from data. What matters in this context is (i) whether we can learn region-based representations of concepts and (ii) whether we can learn vector representations in which dimensions are meaningful and organised into domains. These two issues are discussed in Sections 3.1 and 3.2 respectively.

3.1 Modelling Concepts as Regions

Learning Gaussian Representations In learned vector space embeddings, the objects from some domain of interest are represented as points or vectors, as in conceptual spaces. Most embedding models do not learn region-based representations of concepts. However, if we have access to a number of instances $c_1, ..., c_n$ of a given concept C, we can aim to learn a region-based representation of C from embeddings of these instances. The potential of this strategy stems from the fact that in many embedding models, these instances can be expected to appear clustered together in the vector space. To illustrate this, consider Figure 3, which shows the first two principal components of a 300-dimensional embedding of BabelNet concepts [46] using NASARI vectors[2], which have been learned from Wikipedia and are linked to BabelNet [22]. In the figure, the red points correspond to entities that are instances of the concept Artist, while the blue points correspond to entities that are instances of Painter. For instance, the embeddings of *Edouard Manet*, *Vanessa Bell* and *Claude Monet* appear close to the centre of the blue point cloud. As can be seen, painters appear as a distinct cluster in this vector space embedding.

When attempting to learn a region-based concept representation, we are faced with two challenges: (i) we typically only have access to positive examples and (ii) the number of available instances is often much smaller than the number of dimensions in the vector space. This means that we inevitably have to make some simplifying assumptions to make learning

[2] Downloaded from `http://lcl.uniroma1.it/nasari/`.

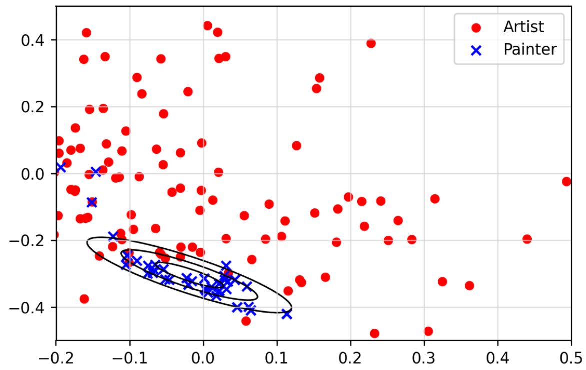


Figure 3 First two principal components of a vector space embedding of BabelNet entities, where blue points correspond to instances of the concept Artist and red points correspond to instances of the concept Painter, according to Wikidata.

possible. A natural choice is to represent concepts as Gaussians. This has the advantage that concept representations can be learned in a principled way, as the problem of estimating Gaussians from observations, either with or without prior knowledge, has been well-studied. Representing concepts using probability distributions, rather than hard regions, also fits well with the view that concept boundaries tend to be fuzzy and ill-defined more often than not. Note that in neural models, concepts are typically represented as vectors, with concept membership determined in terms of dot products, e.g. $\sigma(\mathbf{e} \cdot \mathbf{c})$ is often used to estimate the probability that the entity e (with embedding $\mathbf{e}$) is an instance of concept C (with embedding $\mathbf{c}$), with σ the sigmoid function. This choice effectively means that concepts are represented as spherical regions in the vector space. When using Gaussians, we relax this modelling choice, allowing concepts to be represented using ellipsoidal regions instead.

To deal with the (typically) small number of instances that are available for learning a concept, [17] only considered Gaussians with diagonal covariance matrices. In this case, the problem simplifies to learning a number of univariate Gaussians, i.e. one per dimension. Moreover, a Bayesian formulation with a flat prior was used for estimating the Gaussians. As a consequence, concepts are actually represented using Student t-distributions. The practical implication is that slightly wider ellipsoidal regions are learned than those that would be obtained when using maximum likelihood estimates. Some contours of the learned distribution for the concept Painter are shown in Figure 3.

Bayesian learning with prior knowledge As mentioned above, [17] used a Bayesian formulation for learning Gaussian concept representations. While a flat (i.e. non-informative) prior was used in that paper, the same formulation can be used with informative priors, which offers a natural strategy for incorporating prior knowledge about the concept C being modelled. Such prior knowledge is particularly important when the number of available instances of C is very small (or, in an extreme case, when no instances of C are given at all). This idea was developed in [18], where two sources of prior knowledge were used: ontologies and vector space embeddings of the concept names. In both cases, the prior knowledge allows us to relate the target concept C to other concepts. However, in practice we typically do not yet have a representation of these other concepts, i.e. we are trying to jointly learn a representation of all concepts of interest. This creates circular dependencies, e.g. the representation of concept A provides us with a prior on the representation of concept B, but the representation of concept B also provides us with a prior on the representation of A. This can be addressed using Gibbs sampling; we refer to [18] for the details.

Priors on Mean. Suppose we have concept inclusions of the form $(C \sqsubseteq D_1)$,...,$(C \sqsubseteq D_k)$, and suppose we have a Gaussian representation of the concepts $D_1, ..., D_k$. Then we can induce a prior on the mean of the Gaussian representing C based on the idea that the mean of C should have a high probability in the Gaussians modelling $D_1, ..., D_k$. This can be achieved efficiently by taking advantage of the fact that the product of k Gaussians is proportional to another Gaussian. In addition to ontologies, we can also use vector space embeddings of the (names of the) concepts $C, D_1, ..., D_k$. Specifically, [18] proposed a strategy based on the view that there should be a fixed vector offset between the embedding of a concept C and the mean of the Gaussian that represents it.

Priors on Variance. To obtain a prior on the variance of the Gaussian representing C, we take the view that this variance should be similar to that of the concepts that are most similar to C. To find such concepts, we could take the siblings of C in an ontology, the concepts whose vector space embedding is most similar to the embedding of C, or we could use a hybrid strategy where we select the siblings whose embedding is most similar to that of C. We again refer to [18] for details.

Exploiting contrast sets A common strategy for learning conceptual space representations is to associate each concept with a single point, which intuitively represents its prototype [30]. The region representing a given concept C then consists of all points that are closer to the prototype of C than to the prototype of any other concept, i.e. concept regions are obtained as the Voronoi tessellation of a set of prototype points. This strategy is appealing, because it allows us to learn concept regions with a much wider extension than when learning Gaussians, especially in cases where we only have a few instances per concept. The main idea is illustrated in Figure 4, where we are interested in learning a region for the concept C. When using Gaussians, we would end up with ellipsoidal regions (contours) similar to the ones displayed in the figure. As a result, most points of the space are not assigned to any of the concepts. In contrast, if we construct prototypes by averaging the embeddings of the instances of a concept, and compute the resulting Voronoi tessellation, we essentially carve up the space, as also illustrated in the figure. To see why this can be beneficial in practice, Figure 5 shows the vector representations of the instances of three concepts: Songbook, Brochure and Guidebook. Now consider the left-most test instance of Songbook. If we are only given the training instances of this concept, this test instance is unlikely to be covered by the resulting representation. In contrast, if we instead attempt to carve up the space into regions corresponding to Songbook, Brochure and Guidebook, then this test instance would be classified correctly. The problem with implementing the aforementioned idea is that it only works if we are given a set of concepts that form a *contrast set* [33], i.e. a set of mutually exclusive natural categories that exhaustively cover some sub-domain. For example, the set of all common color names, the set $\{$Fruit, Vegetable$\}$ and the set $\{$NLP, IR, ML$\}$ can intuitively be thought of as contrast sets. We say that two concepts are conceptual neighbours if they belong to the same contrast set and compete for coverage (i.e. are adjacent in the resulting Voronoi tessellation).

Existing ontologies do not typically describe contrast sets or conceptual neighbourhood. To deal with this, [16] introduced a strategy for learning conceptual neighbourhood from data, i.e. for discovering pairs of concepts that are conceptual neighbours. Note that they focus on conceptual neighbourhood rather than contrast sets, as the need for contrast sets to

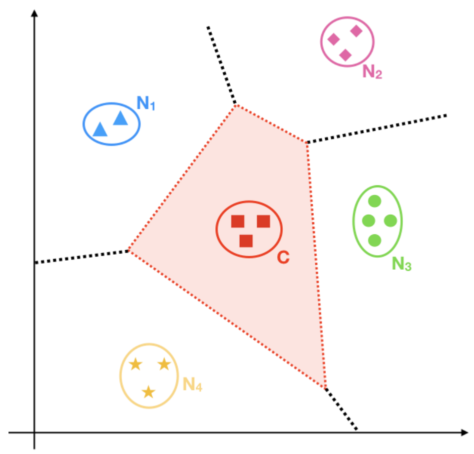


Figure 4 Estimating concept regions based on conceptual neighbourhood.

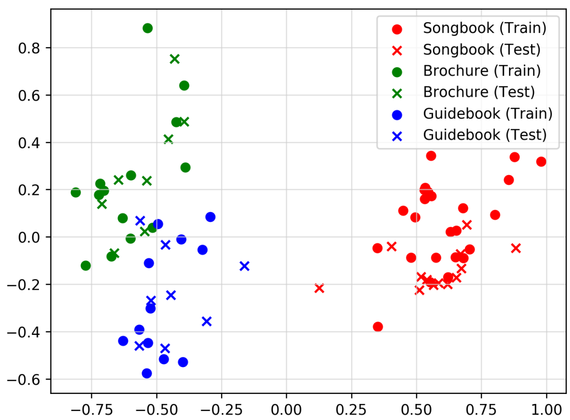


Figure 5 Instances of three BabelNet categories which intuitively can be seen as conceptual neighbors. The figure shows the first two principal components of the NASARI vectors.

be exhaustive is difficult to guarantee. The method then relies on the simplifying assumption that the target concept C, along with its known conceptual neighbours $N_1, ..., N_k$ forms a contrast set. To represent the concept C, first a Gaussian is learned by pooling the instances of $C, N_1, ..., N_k$ together. The ellipsoidal contours of this Gaussian are then carved up into sub-regions for $C, N_1, ..., N_k$ by learning logistic regression classifiers. Specifically, the region representing C is obtained by training logistic regression classifiers that separate the instances of C and N_i, for each $i \in \{1, ..., k\}$. To learn conceptual neighbourhood from data, the first step of the strategy from [16] consists in generating weakly supervised training examples. To this end, they start with two concepts A and B that are siblings in a given taxonomy (i.e. concepts that have the same parent) and for which a sufficiently large number of instances is given. They then compare the performance of the following two types of concept representations:

1. Learn a Gaussian representation of A and B from their given instances.
2. Learn a Gaussian representation from the combined instances of A and B, and then split

High confidence	Medium confidence
Actor – Comedian	Cruise ship – Ocean liner
Journal – Newspaper	Synagogue – Temple
Club – Company	Mountain range – Ridge
Novel – Short story	Child – Man
Tutor – Professor	Monastery – Palace
Museum – Public aquarium	Fairy tale – Short story
Lake – River	Guitarist – Harpsichordist

Table 1 Selected examples of siblings A–B which are predicted to be conceptual neighbours with high and medium confidence.

the resulting region by training a logistic regression classifier that separates A-instances from B-instances.

If the second representations perform (much) better at classifying held-out instances, we can assume that A and B are conceptual neighbours. If the second representations perform much worse, then we can assume that A and B are not conceptual neighbours. In case the performance is similar, then the pair A, B is disregarded when constructing the weakly labelled training set. Table 1 shows some examples of pairs of concepts A, B that were predicted to be conceptual neighbours using this process. Given the resulting training set, we can then train a standard text classifier on sentences that mention both A and B from some text corpus. Consider, for instance, the concepts *Hamlet* and *Village*, and the following sentence [3]:

In British geography, a hamlet is considered smaller than a village and ...

The sentence suggests that *hamlet* and *village* are conceptual neighbors as it makes clear that these concepts are closely related but different. Once a classifier is trained, based on the weakly supervised training set, we can then apply it to other concepts. To learn the representation of a given target concept C (e.g. a concept with only few known instances), we can then use the text classifier to identify which of its siblings, in a given taxonomy, are most likely to be conceptual neighbours, and determine the representation of C accordingly. Tables 2 and 3 show some examples of the top conceptual neighbor predicted by the text classifier, for different target concepts. In particular, Table 3 shows examples where the resulting concept representations (i.e. the representations of the target concepts obtained by exploiting the predicted conceptual neighbourhood) were of high quality, as measured in terms of F1 score for held-out entities. Similarly, Table 2 shows examples where the resulting concept representations were of low quality. As can be seen, the predicted conceptual neighbours in Table 3 are clearly more meaningful than the predicted neighbours in Table 2. This illustrates how the quality of the concept representations is closely linked to our ability to find meaningful conceptual neighbours. Overall, the experiments in [16] showed that using predicted conceptual neighbourhood, on average, led to much better concept representations than when estimating Gaussians from the known instances of the target concept.

3.2 Learning Quality Dimensions

The dimensions of learned vector spaces do not normally correspond to semantically meaningful properties. This is an important difference with conceptual spaces, which severely limits

[3] https://en.wikipedia.org/wiki/Hamlet_(place)

Concept	Top neighbor	F1
Bachelor's degree	Undergraduate degree	34
Episodic video game	Multiplayer gamer	34
501(c) organization	Not-for-profit arts organization	29
Heavy bomber	Triplane	41
Ministry	United States government	33

Table 2 Top conceptual neighbors selected for categories associated with a low F1 score.

Concept	Top neighbor	F1
Amphitheater	Velodrome	67
Proxy server	Application server	61
Ketch	Cutter	74
Quintet	Brass band	67
Sand dune	Drumlin	71

Table 3 Top conceptual neighbors selected for categories associated with a high F1 score.

the interpretability of learned vector space representations. In this section, we review work that has focused on mitigating this issue, by identifying interpretable directions in learned vector spaces. These interpretable directions can then play the role of quality dimensions. This is illustrated in Figure 6, which shows a two-dimensional projection of an embedding of movies from [25]. Along with the embedding of the movies themselves, the figure also shows two directions that have been identified: one direction which ranks the movies from least to most *scary*, and another direction which ranks the movies from least to most *romantic*. Formally, we say that the direction of some vector $\mathbf{v}$ models a property P, such as scary, if for entities e_1 and e_2, with embeddings $\mathbf{e_1}$ and $\mathbf{e_2}$, we have $\mathbf{e_1} \cdot \mathbf{v} > \mathbf{e_2} \cdot \mathbf{v}$ if entity e_1 has the property P to a greater extent than entity e_2.

Identifying quality dimensions Assume that a set of entities $\mathcal{E}$ is given, together with a vector space embedding $\mathbf{e} \in \mathbb{R}^n$ for each entity $e \in \mathcal{E}$. To find interpretable directions, [25] proposed a simple strategy which relies on the assumption that a text description D_e is available for each entity e. Let V be the set of all words (or common multi-word expressions such as "New York") that appear in these descriptions D_e. For $v \in V$, we say that the word v is relevant for the entity e if v appears at least once in the description D_e. It was proposed in [25] to learn a linear classifier in the embedding space, for each $v \in V$, separating the entities for which v is relevant from those for which this is not the case. If this classifier is able to separate these entities well, the assumption is that the word v must be important, i.e. that it describes an aspect that is captured by the embedding space. In this case, the normal vector $\mathbf{v}$ of the hyperplane that was learned by the classifier is treated as a candidate direction. These candidate directions are then clustered, and the each cluster is treated as a quality dimension. This clustering step has the advantage that quality dimensions become easier to interpret, as we have a set of words to describe them, rather than a single word, and it ensures that different quality dimensions are sufficiently different. We refer to [2] for an extensive evaluation of the resulting quality dimensions. We illustrate the main findings with some examples. First, some of the clusters that are found closely correspond to the intuition of quality dimensions. For instance, the following clusters were found in [25], starting from a vector space embedding of movies:

- touching, inspirational, warmth, dignity, sadness, heartwarming, ...

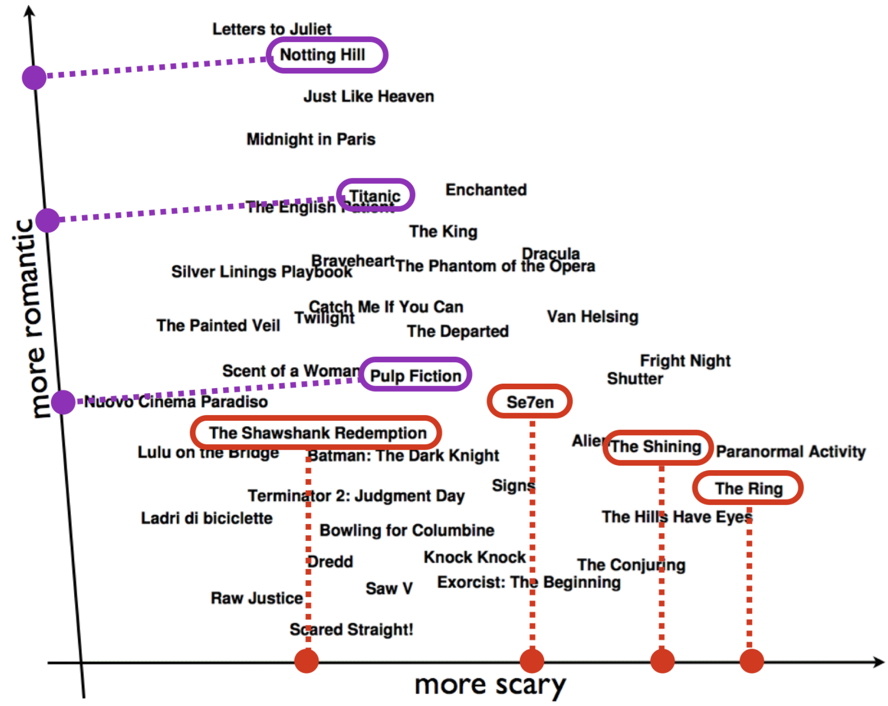


Figure 6 Interpretable directions within a vector space embedding of movies.

- clever, schemes, satire, smart, witty dialogue, ingenious, ...
- bizarre, odd, twisted, peculiar, lunacy, surrealism, obscure, ...
- predictable, forgettable, unoriginal, formulaic, implausible, contrived, ...
- tragic, anguish, sorrow, fatal, misery, bitter, heartbreaking, ...
- romantic, lovers, romance, the chemistry, kisses, true love, ...
- eerie, paranoid, spooky, impending doom, dread, ominous, ...
- scary, shivers, chills, creeps, frightening, the dark, goosebumps, ...
- cheesy, camp, corny, tacky, laughable, a guilty pleasure, ...
- hilarious, humorous, really funny, a very funny movie, amusing, ...
- wonderful, fabulous, a joy, gem, delighted, happy, perfect, great, ...

Arguably, all these directions correspond to clear and salient semantic attributes of movies. On the other hand, many other clusters rather corresponded to movie themes, e.g.:

- horror movies, zombie, much gore, slashers, vampires, scary monsters, ...
- killer, stabbings, a psychopath, serial killer, ...
- supernatural, a witch, ghost stories, mysticism, a demon, the afterlife, ...
- scientist, experiment, the virus, radiation, the mad scientist, ...
- criminal, the mafia, robbers, parole, the thieves, the mastermind, ...

While these directions express semantically meaningful properties, it would be more natural to represent such properties as regions than as quality dimensions. The fact that such thematic properties cannot be distinguished from the semantic attributes mentioned above is clearly a limitation of the method from [25]. In [2], it was found that the nature of the clusters, i.e. whether they intuitively correspond to quality dimensions rather than thematic properties, to some extent depends on the scoring function that is used for evaluating the linear classifiers. However, regardless of the scoring function that is used, a mixture of different types of properties is found. One possible solution could be to require that clusters which correspond to quality dimensions should contain a sufficient proportion of adjectives, as clusters consisting mostly of nouns are more likely to be thematic properties. On the other hand, it is not clear that having thematic "quality dimensions" is necessarily problematic. While it makes the resulting representation different from a conceptual space, it still allows us to "disentangle" the vector representation into different aspects (e.g. genre, sentiment, emotion). Furthermore, a cluster of terms related to horror movies could still be viewed as a quality dimension if we view it as ranking movies based on how "horror-like" they are.

A number of improvements to the basic method from [25] have been explored. In [3] a fine-tuning strategy is introduced, which modifies the initial vector space based on the discovered quality dimensions, while [6] suggests to learn quality dimensions in a hierarchical fashion, with the top-level dimensions essentially partitioning the vector space into thematic domains, and the lower-level dimensions intuitively corresponding to quality dimensions within each of these thematic domains. In terms of how the resulting quality dimensions could be useful, the main focus has so far been on their ability to support interpretable classifiers, with [25] introducing a rule based classifier, which compares entities with training examples along a small number of quality dimensions, and [3, 6] using the quality dimensions as features for low-depth decision trees.

Organising quality dimensions into domains The quality dimensions of a conceptual space are organised into domains. Accordingly, as we have seen in the previous section, the quality dimensions that can be identified in learned vector spaces also intuitively belong to different kinds. It would be of interest to group quality dimensions of the same kind together, to learn a structure which is akin to conceptual space domains. For instance, in the movies domain, we could imagine one group of quality dimensions about the emotion a movie evokes, as well as groups about the genre, the cinematographic style, etc. We will refer to these groups of learned quality dimensions as *facets*, rather than domains, to avoid confusion (e.g. domain can also refer to the domain-of-discourse, such as movies, or to the domain of a description logic interpretation) and to highlight the fact that there are still important differences between these facets and conceptual space domains. In addition to grouping quality dimensions that are concerned with the same aspect of meaning, we also want to learn a corresponding lower-dimensional vector space for each facet. In other words, the central aim is to decompose the given vector space into a number of lower-dimensional spaces, each of which captures a different aspect of meaning.

Note that we cannot learn these facets by simply clustering the quality dimensions. For instance, *thriller* and *scary* may be represented by similar directions in the vector space, but they should be assigned to different facets. In contrast, *romance* and *horror* would be represented by dissimilar directions but nonetheless belong to the same facet. The key solution, which was developed in [5] and [4], is to rely on word embeddings to identify words that describe properties of the same kind. For instance, the word embeddings of different movie genres tend to be similar, because such words tend to appear in similar contexts. In the same way, different adjectives describing emotions tend to be represented using similar word vectors. This suggests a simple strategy for learning facets: (i) cluster the word vectors of the words associated with the quality dimensions that were identified in the given vector space; and (ii) represent the facet by the vector space that is spanned by quality dimensions that are assigned to it. Unfortunately, this strategy was found to perform poorly in [5]. The main reason is that in many areas there is one dominant facet, such as the genre in the case of movies. When applying the aforementioned strategy, what happens is that each of the resulting facet-specific vector spaces mostly models the dominant facet. To address this issue, [5] proposed an iterative strategy, in which the dominant facet is first identified and then explicitly disregarded when determining the second facet, etc. Another practical challenge is that the overall method is computationally demanding, especially the fact that a linear classifier has to be learned for each word from the vocabulary, to identify the interpretable directions (in the overall space and in each of the lower-dimensional facet-specific spaces). To address this issue, [4] introduced a model that directly learns facet-specific vector spaces from bag-of-words representations of the entities, using a mixture-of-experts model to generalize the GloVe [53] word embedding model. Using this approach, facet-specific vector spaces can be learned much more efficiently, and moreover the resulting embeddings tend to be of a higher quality. The main limitation, however, is that this model assumes that suitable vector spaces can be learned from bag-of-words representations (rather than being agnostic to how the initial vector space embedding is learned) and that GloVe is a suitable embedding model for learning these vector spaces.

The resulting facet-specific embeddings can be used in a number of different ways. Perhaps the most immediate application of such representations is that they facilitate concept learning. For instance, suppose we want to represent each concept as a Gaussian. Furthermore, suppose that only one of the facet-specific vector spaces is relevant for modelling the considered concept. If we learn a Gaussian in each of the factor-specific vector spaces, we should end up with Gaussian with a large variance for the irrelevant facets, and a Gaussian with a much lower variance in the vector space corresponding to the relevant facet. This advantage of facet-specific vector spaces was empirically confirmed in [4]. Moreover, they found that even strategies that only rely on the resulting quality dimensions, e.g. learning low-depth decision trees, were benefiting from learning facet-specific vector spaces, as the lower-dimensional nature of each vector space acts as a regulariser.

4 Modelling Relations with Conceptual Spaces

Conceptual spaces act as an interface between vector space embeddings and symbolic knowledge. However, because conceptual spaces do not capture relational knowledge, they are essentially limited to capturing Horn rules with unary predicates. In this section, we explore whether the framework of conceptual spaces can be generalised to encode rules with binary and higher arity relations. We focus on the analysis presented in [37] but use a construction that is somewhat more intuitive than the one used in the latter paper. The main idea is to view a k-ary relation as a convex region in the Cartesian product of k conceptual spaces. For simplicity, in this section we will assume that conceptual spaces correspond to Euclidean spaces. Each individual a is then represented as a vector $\mathbf{a} \in \mathbb{R}^n$. A tuple $(a_1, ..., a_k)$ is represented as the concatenation of the vectors representing $a_1, ..., a_k$, i.e. $(a_1, ..., a_k)$ is represented as the $n \cdot k$-dimensional vector $\mathbf{a_1} \oplus ... \oplus \mathbf{a_k}$, where we write $\oplus$ for vector concatenation.

The main idea is illustrated in Figure 7. In this toy example, we assume that individuals are represented in a one-dimensional conceptual space. Unary predicates such as herbivore then correspond to intervals, while binary predicates such as eats correspond to convex regions in $\mathbb{R}^2$. In this figure, the tuple $(\mathsf{lion}, \mathsf{zebra})$ corresponds to a point in the region encoding the eats predicate. This captures the knowledge that lions eat zebras. Moreover, we can now also model dependencies between unary and binary predicates. For instance, the

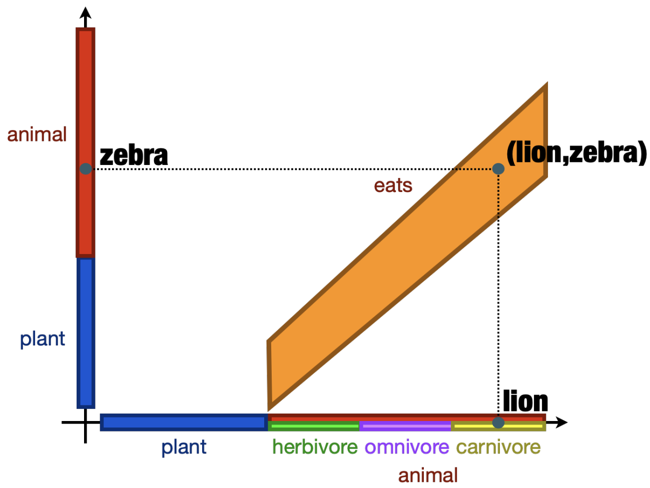


Figure 7 Illustration of a relational conceptual space.

representation captures the following rule:

$$\mathsf{carnivore}(X) \leftarrow \mathsf{eats}(X,Y), \mathsf{animal}(Y)$$

This can be seen as follows. Consider a point $\mathbf{p} \in \mathbb{R}^2$ in the region representing eats, such that its projection on the Y-axis lies in the interval representing animal. For each such a point $\mathbf{p}$, it holds that its projection on the X-axis lies in the interval representing carnivore. We can think of each point $\mathbf{p}$ as the representation of a possible instantiation of the tuple (X,Y). The aforementioned observation about $\mathbf{p}$ then corresponds to the view that every tuple satisfying the body of the rule also satisfies its head. In a similar way, we can also model rules with existential quantifiers, e.g.:

$$\exists Y.\mathsf{eats}(X,Y) \wedge \mathsf{animal}(Y) \leftarrow \mathsf{carnivore}(X)$$

To see why this rule is satisfied for the configuration depicted in Figure 7, consider a value $x \in \mathbb{R}$ which lies in the interval representing carnivore. Then we can always find a coordinate $y \in \mathbb{R}$ such that the point $\mathbf{p} = (x,y)$ lies in the region for eats and such that y lies in the interval modelling animal. In Section 4.1 we discuss these intuitions in more detail. We also provide a characterisation about the kinds of relational rules that can be modelled using convex regions. Subsequently, in Section 4.2 we discuss the relationship with knowledge graph embedding models.

4.1 Geometric Models of Relational Rules

We consider geometric interpretations η, which map each individual a to a point $\eta(a) \in \mathbb{R}^n$ and each k-ary relation r to a convex region $\eta(r)$ in $\mathbb{R}^{k\cdot n}$. These geometric interpretations can intuitively be seen as defining a relational counterpart to conceptual spaces. We now discuss what it means for a geometric interpretation η to satisfy different kinds of relational knowledge. First, a relational fact of the form $r(a_1,...,a_k)$ is satisfied if the representation of the tuple $(a_1,...,a_k)$ lies in the region representing r, i.e.:

$$\eta(a_1) \oplus ... \oplus \eta(a_k) \in \eta(r)$$

Now we consider a basic relational entailment of the following form:

$$r(X_1, ..., X_k) \leftarrow s(X_1, ..., X_k)$$

This rule is satisfied if the region modelling s is included in the region modelling r, i.e. it corresponds to the following geometric constraint:

$$\eta(s) \subseteq \eta(r)$$

Conjunctions in the body of a rule can be modelled using intersections. For instance, consider the following rule:

$$r(X_1, ..., X_k) \leftarrow s(X_1, ..., X_k), t(X_1, ..., X_k) \tag{8}$$

The corresponding geometric constraint is as follows:

$$\eta(s) \cap \eta(t) \subseteq \eta(r)$$

This simple geometric characterisation only works because each relation is applied to the same tuple $(X_1, ..., X_k)$. To see how we can model more general rules, let us consider a rule of the following form:

$$r(X, Z) \leftarrow s(X, Y), t(Y, Z) \tag{9}$$

The main idea is to view this rule as a special case of (8). In particular, let us consider ternary relations r^*, s^* and t^* defined as follows: $r^*(X, Y, Z) \equiv r(X, Z)$, $s^*(X, Y, Z) \equiv s(X, Y)$ and $t^*(X, Y, Z) \equiv t(Y, Z)$. Then clearly (9) is equivalent to:

$$r^*(X, Y, Z) \leftarrow s^*(X, Y, Z), t^*(X, Y, Z)$$

whose geometric characterisation is given by $\eta(s^*) \cap \eta(t^*) \subseteq \eta(r^*)$. This is illustrated in Figure 8, where the relationship between the two-dimensional regions $\eta(r)$, $\eta(s)$, $\eta(t)$ and the three-dimensional regions $\eta(r^*)$, $\eta(s^*)$, $\eta(t^*)$ is shown. To explain how the regions $\eta(r^*)$, $\eta(s^*)$, $\eta(t^*)$ relate to $\eta(r)$, $\eta(s)$, $\eta(t)$ more formally, we have to introduce some notations. Let $I = \{i_1, ..., i_l\} \subseteq \{1, ..., k\}$ be a set of indices. For a point $(x_1, ..., x_{k\cdot n}) \in \mathbb{R}^{k\cdot n}$, we define its *restriction to* I as follows

$$(x_1, ..., x_{k\cdot n}) \downarrow I = \bigoplus_{i \in I} (x_{n\cdot i+1}, ..., x_{n\cdot i+n})$$

For instance if $n = 2$, $k = 4$ and $I = \{1, 4\}$ we have $(x_1, ..., x_8) \downarrow I = (x_1, x_2, x_7, x_8)$. In particular, note that when $(x_1, ..., x_{k\cdot n})$ is the representation of a tuple $(a_1, ..., a_k)$, and $(b_1, ..., b_l)$ is obtained from $(a_1, ..., a_k)$ be only keeping the arguments at the positions in I, then $\eta(b_1, ..., b_l) = \eta(a_1, ..., a_k) \downarrow I$. We define the notion of *cylindrical extension* as follows. Let R be a region in $\mathbb{R}^{l\cdot n}$ with $l < k$ and let $I = \{i_1, ..., i_l\} \subseteq \{1, ..., k\}$ Then we define:

$$\mathsf{ext}_I^k(R) = \{\mathbf{x} \in \mathbb{R}^{k\cdot n} \,|\, \mathbf{x} \downarrow I \in R\}$$

Let us now return to the problem of modelling the rule (9). We have $\eta(r^*) = \mathsf{ext}^3_{\{1,3\}}(\eta(r))$, $\eta(s^*) = \mathsf{ext}^3_{\{1,2\}}(\eta(s))$ and $\eta(t^*) = \mathsf{ext}^3_{\{2,3\}}(\eta(t))$. We thus find that the rule (9) corresponds to the following geometric constraint:

$$\mathsf{ext}^3_{\{1,2\}}\big(\eta(s)\big) \cap \mathsf{ext}^3_{\{2,3\}}\big(\eta(t)\big) \subseteq \mathsf{ext}^3_{\{1,3\}}\big(\eta(r)\big)$$

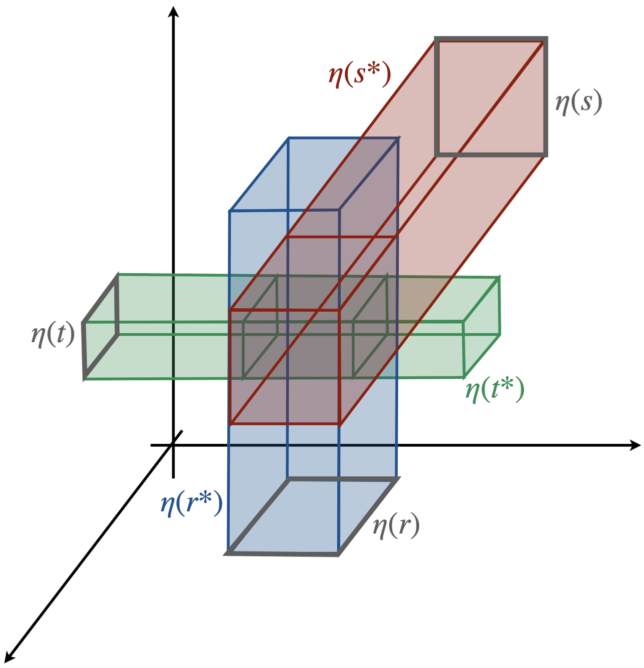


Figure 8 Illustration of the constraint $\eta(s^*) \cap \eta(t^*) \subseteq \eta(r^*)$.

While the rule (9) only involves binary relations, clearly we can apply the same strategy to rules involving relations of other arities, and to rules with more than two atoms in the body. Finally, we discuss how rules with existential quantifiers can be modelled. Let us consider the following example:

$$\exists Y \,.\, r(X,Y) \wedge s(Y,Z) \leftarrow t(X,Z) \tag{10}$$

The key challenge is to characterise the region that models the head of this rule. Note that, as before, $r(X,Y) \wedge s(Y,Z)$ can be modelled by treating r and s as ternary relations. Relying again on the cylindrical extension, we find that this conjunction can be modelled as $\mathsf{ext}^3_{\{1,2\}}(\eta(r)) \cap \mathsf{ext}^3_{\{2,3\}}(\eta(s))$. To model the existential quantifier, we can then simply remove the coordinates pertaining to the variable Y. In other words, the rule (10) corresponds to the following geometric constraint:

$$\eta(t) \subseteq \left(\mathsf{ext}^3_{\{1,2\}}\big(\eta(r)\big) \cap \mathsf{ext}^3_{\{2,3\}}\big(\eta(s)\big)\right) \downarrow \{1,3\}$$

In this way, using a combination of cylindrical extensions and projections, any relational rule can be translated into a corresponding geometric constraint. It is worth pointing out that a similar treatment of rules was already proposed by Zadeh [78] in his theory of approximate reasoning. The main difference with the aforementioned approach is that relations in the latter case are modelled as fuzzy sets.

A central question is which kinds of rules can be faithfully[4] modelled in terms of the aforementioned geometric constraints. The answer depends on which kinds of regions we allow as the geometric interpretation $\eta(r)$ of a relation r. Without any restrictions, arbitrary

[4] Note that we use this notion of faithfulness informally here; we refer to [37] for a formal treatment of geometric models.

sets of relational rules can be modelled correctly. However, in practice, it makes sense to require $\eta(r)$ to be convex. While the cognitive plausibility of this assumption is unclear, in practice we can only hope to learn region-based representations in high-dimensional spaces by making drastic simplifying assumptions, as we also saw in Section 3. For this reason, most strategies for modelling relational knowledge end up learning convex representations; this will be discussed in more detail in Section 4.2. With this convexity assumption, however, clearly some sets of rules cannot be jointly modelled. For instance we cannot model the rule $\bot \leftarrow r_1(X,Y), r_2(X,Y)$, capturing that relations r_1 and r_2 are disjoint, together with the following facts: $r_1(a,a)$, $r_1(b,b)$, $r_2(a,b)$, $r_2(b,a)$. Indeed, if $\eta(r_1)$ and $\eta(r_2)$ are convex, from $\eta(a) \oplus \eta(a) \in \eta(r_1)$, $\eta(b) \oplus \eta(b) \in \eta(r_1)$, $\eta(a) \oplus \eta(b) \in \eta(r_2)$ and $\eta(b) \oplus \eta(a) \in \eta(r_2)$, we find:

$$\frac{(\eta(a)+\eta(b))}{2} \oplus \frac{(\eta(a)+\eta(b))}{2} \in \eta(r_1) \cap \eta(r_2)$$

and thus r_1 and r_2 are not disjoint in the geometric interpretation η. However, in [37] it was shown that many sets of relational rules can still be faithfully captured by geometric models. In particular, consider a relational rule of the following form:

$$\exists Y_1, ..., Y_r . H_1 \wedge ... \wedge H_s \leftarrow B_1, ..., B_t$$

where $H_1, ..., H_s, B_1, ..., B_t$ are atoms. We say that such a rule is quasi-chained, if every atom B_i appearing in the body shares at most 1 variable with the atoms $B_1, ..., B_{i-1}$. It can be shown that any set of quasi-chained rules with a finite model can be faithfully captured by a geometric model in which every relation is represented as a convex region [37]. Some open questions remain, however, including the following:

- Is there a larger fragment of existential rules that can be faithfully modelled using geometric interpretations with convex regions?
- Is there a way to relax the convexity assumption, such that arbitrary existential rules can be captured, while keeping representations simple enough to be learnable?

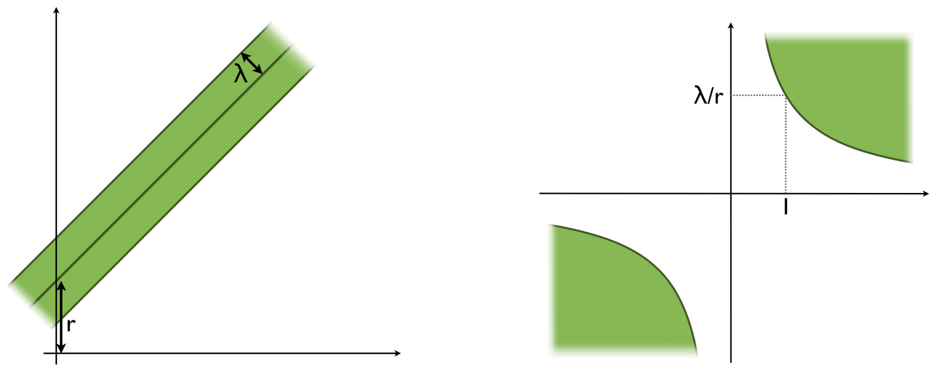
Finally, it should be noted that the restriction to arbitrary convex regions means that negation and disjunction cannot easily be modelled. Some authors have proposed geometric models that were specifically designed with such logical connectives in mind, including the use of axis aligned cones [52]. Recently, the ability of convex regions to model temporally attributed description logics has also been studied [19].

4.2 Link with Knowledge Graph Embedding

Thus far, we have not discussed how region-based representations of relations may be learned from data. In the last few years, there has been an increasing interest in region based representations, as already mentioned in Section 3.1. Most approaches, however, only use regions for modelling concepts, and deal with relations in an ad hoc way. For instance, the approach from [79] represents entities using cones, but uses a feed-forward neural network for modelling relations. Similarly, [52] propose a cone based model for embedding $\mathcal{ALC}$ ontologies, but they refrain from modelling roles in the same way. However, in [1] a knowledge graph embedding model is proposed in which relations are explicitly modelled as hyperboxes. More generally, many of the standard knowledge graph embedding models can be interpreted as region based models. In particular, for a relation r with scoring function f_r we can consider the following region:

$$\eta(r) = \{\mathbf{e} \oplus \mathbf{f} \mid f_r(e,f) \geq \lambda_r\}$$

with λ_r some threshold. Figure 9 illustrates how TransE and DistMult can be viewed as region-based models in this way. However, viewed as region based models, TransE and



(a) TransE **(b)** DistMult

Figure 9 Region based view of knowledge graph embedding models.

bilinear models such as DistMult are severely limited in which kinds of existential rules they can capture; we refer to [37] for more details.

5 Plausible Symbolic Reasoning using Vector Space Embeddings

Leaving aside the difficulties of tightly integrating geometric and symbolic representations, it is highly relevant for the development of robust AI systems to understand how symbolic approaches to AI can be made more flexible by equipping them with inductive capabilities, i.e. making it possible to infer likely concept inclusions (or rules) by using the knowledge of the ontology in combination with the additional background knowledge provided by vector representations. In other words, one would like symbolic systems to incorporate mechanisms to use predictions made by neural approaches, informing about plausible situations witnessed in the data, in a principled way. In the rest of this section we will discuss ways in which this idea can be implemented.

One of the most natural solutions is to use vector representations to implement a form of similarity based reasoning [23, 13]. For instance, we could have a KB with factual knowledge stating that strawberries are instances of the concept berries, Berry(strawberry), and ontological knowledge stating that berries are healthy, Berry $\sqsubseteq$ Healthy. Clearly, this KB entails that strawberries are healthy. Further, using a standard word embedding [45], we can find out that strawberry and raspberry are highly similar. Now, using the KB and the additional similarity information, we can infer that it is plausible that raspberries are berries and, therefore, healthy. This same idea could be lifted to find the similarity between concept names (classes) and find plausible rules. For instance, assume that strawberries and raspberries are concept names and that our ontology specifies that strawberries are healthy, i.e. Strawberry $\sqsubseteq$ Healthy. Using the similarity between strawberries and raspberries, we could then infer that the concept inclusion Raspberry $\sqsubseteq$ Healthy is plausible. However, implementing this strategy in a principled way is difficult, because it is unclear how we can formally relate degrees of similarity to the plausibility of the inferred rules, i.e. if we can infer using standard deduction that $C_1 \sqsubseteq X$, how similar does concept C_2 needs to be to C_1 to accept the plausible inference $C_2 \sqsubseteq X$? For this reason, rather than focusing on similarity based reasoning, it has been proposed to focus on *interpolative reasoning* instead [64]. The main difference is that instead of focusing on the similarity between two entities, we focus on how one entity relates to a group of entities. For instance, we say that the concept

Raspberry is *conceptually between* the concepts Strawberry, Blackberry and Cherry. Intuitively, this means that we accept that any (natural) property that holds for each of the concepts Strawberry, Blackberry, Cherry is likely to hold for Raspberry as well. In addition to using similarity based strategies, humans also rely on analogies for inferring plausible knowledge. Analogical reasoning can be particularly powerful, as it allow us to make predictions about concepts that may themselves not be similar to any other concepts. Recent models from the field of Natural Language Processing make it possible to discover analogies with a high level of accuracy [71]. It is thus of interest to explore whether analogy based reasoning processes could be used as another mechanism for exploiting knowledge from neural representations for symbolic reasoning. We now discuss in more detail how interpolative and analogical reasoning can be formalised in the context of description logics.

5.1 Interpolative Reasoning

We start by illustrating how the interpolation pattern works [26, 64]. Assume that we have the following knowledge about some concept C:

$$\mathsf{Strawberry} \sqsubseteq C \qquad \mathsf{Orange} \sqsubseteq C$$

Intuitively, even if we know nothing else about C, we could still make the following inductive inference:

$$\mathsf{Raspberry} \sqsubseteq C \tag{11}$$

This conclusion relies on background knowledge about strawberries, oranges and raspberries, in particular the fact that raspberries are expected to have all the *natural* properties that strawberries and oranges have in common (e.g. being high in vitamin C). In such a case, we say that raspberries are *conceptually between* strawberries and oranges. Importantly, knowledge about conceptual betweenness can be derived from data-driven representations. For instance, [25] found that geometric betweenness closely corresponds to conceptual betweenness in vector spaces learned with multi-dimensional scaling.

The notion of *naturalness* plays a central role, as it is clear that the conclusion in (11) can only be justified by making certain assumptions on the concept C. If C could be an arbitrary concept, e.g. a concept representing the union of Orange and Strawberry, there is no reason to believe that the inference is valid, but for natural properties the inference seems intuitively plausible. This idea that only some properties admit inductive inferences has been extensively studied in philosophy [34, 57, 27]. In the context of conceptual spaces, "natural properties" are those which are modelled as convex regions, as explained in Section 2.1. To determine which concepts, in a given ontology, are likely to be natural, a useful heuristic is to consider the concept name: concepts that correspond to standard natural language terms are normally assumed to be natural [29].

The extension $\mathcal{EL}^{\bowtie}$ of $\mathcal{EL}$ was designed based on the above intuitions, with the aim of enabling reasoning about conceptual betweenness and natural concepts, and thus supporting interpolative reasoning. Syntactically, $\mathcal{EL}$ is extended with the in-between constructor, which allows us to describe the set of objects that are between two concepts: we write $C \bowtie D$ to denote all objects that are between the concepts C and D. We further assume that $\mathsf{N_C}$ contains a distinguished infinite set of *natural concept names* $\mathsf{N_C^{Nat}}$. The syntax of $\mathcal{EL}^{\bowtie}$ *concepts* C, D is thus defined by the following grammar, where $A \in \mathsf{N_C}$, $A' \in \mathsf{N_C^{Nat}}$ and $r \in \mathsf{N_R}$:

$$C, D := \top \mid A \mid C \sqcap D \mid \exists r.C \mid N \qquad N, N' := A' \mid N \sqcap N' \mid N \bowtie N'$$

Concepts of the form N, N' are called *natural concepts*.

▶ **Example 4.** Using the following $\mathcal{EL}^{\bowtie}$ TBox $\mathcal{T}$, we can now model the situation described above:

$$\mathsf{Strawberry} \sqsubseteq \mathsf{Healthy} \tag{12}$$

$$\mathsf{Orange} \sqsubseteq \mathsf{Healthy} \tag{13}$$

$$\mathsf{Raspberry} \sqsubseteq \mathsf{Strawberry} \bowtie \mathsf{Orange} \tag{14}$$

$$\mathsf{Healthy} \sqsubseteq \exists \mathsf{improves.QualityOfLife} \tag{15}$$

such that $\mathsf{Strawberry}, \mathsf{Orange}, \mathsf{Raspberry}, \mathsf{Healthy} \in \mathsf{N}_\mathsf{C}^\mathsf{Nat}$.

The semantics of $\mathcal{EL}^{\bowtie}$ needs to adequately characterise natural concepts and concept betweenness, and thus support interpolation, i.e.: such that from $A \sqsubseteq B_1 \bowtie B_2$, $B_1 \sqsubseteq C$ and $B_2 \sqsubseteq C$, we can derive $A \sqsubseteq C$, provided that C is *natural*. To this end, Ibáñez-García et al. [41] proposed two semantics: a feature-enriched semantics inspired by formal concept analysis [73] and a geometric semantics inspired by conceptual spaces. In the former, at the semantic level a set of features is associated with each concept. Note that these features are semantic constructs, which have no direct counterpart at the syntactic level. A concept is then natural if it is completely characterized by these features, while B is between A and C if the set of features associated with B contains the intersection of the sets associated with A and C. In the second semantics, concepts are interpreted as regions from a vector space. A concept is then natural if it is interpreted as a convex region, while B is between A and C if the region corresponding to B is geometrically between the regions corresponding to A and C (i.e. in the convex hull of their union). We refrain from giving the full technical details, but invite the interested reader to look at [41]. Ibáñez-García et al. [41] also investigate the complexity of reasoning with interpolation, and show that under both semantics the concept subsumption problem becomes computationally more costly than in pure $\mathcal{EL}$: coNP-complete under the feature semantics and PSpace-hard under the geometric semantics.

One of the main drawbacks of the feature semantics is that it is too restrictive and cannot support interpolation in an adequate way if the $\bot$ construct is present. To address this shortcoming, Schockaert et al. [62] recently introduced a new semantics based on an abstract ternary betweenness relation bet over elements of the domain, such that that $\mathsf{bet}(a, b, c)$ if b is between a and c. We then have that $A \sqsubseteq B_1 \bowtie B_2$ is satisfied in an interpretation $\mathcal{I}$ if every element in $A^\mathcal{I}$ is between some individual from $B_1^\mathcal{I}$ and some element from $B_2^\mathcal{I}$. A central result from [62] shows that the feature-enriched semantics from [41] can essentially be seen as a special case, where the betweenness relation bet fulfills certain properties. The results in [62] are preliminary, leaving open for example, the complexity of reasoning under this new semantics.

The logic $\mathcal{EL}^{\bowtie}$ is built on the idea of conceptual betweenness. This ensures that the semantics remains close to cognitive models of category based induction, and information about conceptual betweenness can moreover be readily obtained from embeddings. However, an important open question is whether it is possible to develop meaningful forms of rule interpolation that go beyond this idea of conceptual betweenness. For instance, consider the following rules:

$$\mathsf{burglary}(L,T) \leftarrow \mathsf{burglary}(L,T-2), \mathsf{burglary}(L,T-1)$$
$$\mathsf{burglary}(L,T) \leftarrow \mathsf{burglary}(L,T-1), \mathsf{burglary}(L_1,T-1), \mathsf{burglary}(L_2,T-1), \mathsf{n}(L,L_1), \mathsf{n}(L,L_2), L_1 \neq L_2$$
$$\mathsf{burglary}(L,T) \leftarrow \mathsf{burglary}(L,T-2), \mathsf{burglary}(L_1,T-1), \mathsf{burglary}(L_2,T-1), \mathsf{n}(L,L_1), \mathsf{n}(L,L_2), L_1 \neq L_2$$

Intuitively, these rules partially characterise the spatio-temporal diffusion pattern of crime hotspots. For instance, the first rule asserts that if there has been a burglary at time points $T-1$ and $T-2$ at a given location (e.g. during the two previous days), then it is likely that there will be a burglary at time point T in the same location. The other two rules include the predicate n to encode information about neighbouring locations. Given these rules, the following rule also seems plausible:

$$\mathsf{burglary}(L,T) \leftarrow \mathsf{burglary}(L_1,T-2), \mathsf{burglary}(L_2,T-2), \mathsf{burglary}(L,T-1), \mathsf{n}(L,L_1), \mathsf{n}(L,L_2), L_1 \neq L_2$$

However, it is unclear how the underlying principle could be formalised, and how the associated background information could be obtained.

5.2 Analogical Reasoning

Reasoning by analogy has been extensively studied in cognitive science, philosophy, and artificial intelligence [31, 38, 39, 12, 55, 11]. In the context of AI, the formalisation of analogical reasoning typically builds on analogical proportions, i.e. statements of the form *"A is to B what C is to D"* [12, 55, 11]. For instance, a notable result in this area has been the development of analogical classifiers, which are based on the principle that whenever the features of four examples are in an analogical proportion, then their class labels should be in an analogical proportion as well [12, 40]. Somewhat surprisingly, analogical reasoning was only recently considered for completing ontologies [61]. Schockaert et al. [61] took inspiration from analogical classifiers to infer plausible concept inclusions. The resulting inference pattern is called *rule extrapolation*; it is illustrated in the next example.

▶ **Example 5** ([61], Rule Extrapolation). Suppose we have an ontology with the following concept inclusions:

$$\mathsf{Young} \sqcap \mathsf{Cat} \sqsubseteq \mathsf{Cute} \tag{16}$$
$$\mathsf{Adult} \sqcap \mathsf{WildCat} \sqsubseteq \mathsf{Dangerous} \tag{17}$$
$$\mathsf{Young} \sqcap \mathsf{Dog} \sqsubseteq \mathsf{Cute} \tag{18}$$

Suppose we are furthermore given that "Cat is to WildCat what Dog is to Wolf". Trivially, we also have that "Young is to Adult what Young is to Adult" and "Cute is to Dangerous what Cute is to Dangerous". Using rule extrapolation, we can then infer the following:

$$\mathsf{Adult} \sqcap \mathsf{Wolf} \sqsubseteq \mathsf{Dangerous} \tag{19}$$

The knowledge inferred by analogical reasoning could also be used to transfer knowledge from one domain to another:

▶ **Example 6** ([61], Rule translation)**.** Suppose we are given the following knowledge:

$$\mathsf{Program} \sqsubseteq \exists \mathsf{specifies}.\mathsf{Software} \tag{20}$$

and the fact that "Program is to Plan what Software is to Building". Then we can plausibly infer:

$$\mathsf{Plan} \sqsubseteq \exists \mathsf{specifies}.\mathsf{Building} \tag{21}$$

Rule translation is useful as ontologies are often developed using "templates" to encode knowledge from different domains (e.g. knowledge about different professions). The strategy from Example 6 then allows us to complete the ontology by introducing additional domains.

As in the case of interpolative reasoning, the main objective of Schockaert et al. [61] was to establish the principles for incorporating analogical reasoning and, in particular, to develop a model-theoretic semantics. To this end, the description logic $\mathcal{EL}_\perp^{\mathsf{ana}}$ is introduced, which extends $\mathcal{EL}_\perp^{\bowtie}$ with so-called analogy assertions. Formally, $\mathcal{EL}_\perp^{\mathsf{ana}}$ concepts C, D are defined by the following grammar, where $A \in \mathsf{N_C}$, $A' \in \mathsf{N_C^{Nat}}$, $r \in \mathsf{N_R}$ and $r' \in \mathsf{N_R^{Int}}$:

$$C, D := \top \mid \perp \mid A \mid C \sqcap D \mid \exists r.C \mid N$$
$$N, N' := A' \mid N \sqcap N' \mid N \bowtie N' \mid \exists r'.N$$

Note how $\mathcal{EL}_\perp^{\mathsf{ana}}$ concepts extend $\mathcal{EL}_\perp^{\bowtie}$ concepts by allowing existential restrictions over so-called intra-domain roles, i.e. roles from the designated set $\mathsf{N_R^{Int}}$, as natural concepts. An $\mathcal{EL}_\perp^{\mathsf{ana}}$ TBox is a finite set containing two types of assertions: (i) $\mathcal{EL}_\perp^{\mathsf{ana}}$ concept inclusions, and (ii) *analogy assertions* of the form $C_1 \triangleright D_1 :: C_2 \triangleright D_2$, where C_1, C_2, D_1, D_2 are natural $\mathcal{EL}_\perp^{\mathsf{ana}}$ concepts.

The semantics of $\mathcal{EL}_\perp^{\mathsf{ana}}$ builds on the feature-enriched semantics of $\mathcal{EL}_\perp^{\bowtie}$. Recall that analogies involve transferring knowledge from one application domain to another domain, e.g. from software engineering to architecture. Hence, at the semantic level these domains will be associated with subsets of features $\mathcal{F}$. In particular, interpretations will specify a partition $[\mathcal{F}_1, ..., \mathcal{F}_k]$ of $\mathcal{F}$, defining the different domains of interest. To capture the intuition of analogies, some of the partition classes will be viewed as being analogous, in the sense that there is some kind of structure-preserving mapping between them. We again refrain from giving the full technical details. We point out that Schockaert et al. [61] formally show that the analogical patterns exemplified above are supported under the proposed semantics.

The investigation by Schockaert et al. [61] leaves open several interesting questions such as establishing the computational complexity of reasoning in $\mathcal{EL}_\perp^{\mathsf{ana}}$. For the practical uptake of $\mathcal{EL}_\perp^{\mathsf{ana}}$, it would be also important to consider nonmonotonic extensions, as analogical assertions might introduce conflicts with the existing ontological knowledge.

6 Conclusions

Combining symbolic reasoning with sub-symbolic learning is an important and widely studied challenge for AI research. To enable such a combination in a principled way, a key question is how we can unify the two rather distinct types of representations that are involved, i.e. symbols and vectors. In this paper, we discussed a number of strategies that are inspired by the theory of conceptual spaces. First, we looked at the possibility of achieving a tight integration between symbolic and vector representations based on the idea that concepts can be viewed as regions in vector space embeddings. Moreover, we also explored the idea that meaningful "quality dimensions" can be identified in learned embeddings, adding more structure and a degree of interpretability to the vector representations themselves. However, we also argued that the use of region based representations has some inherent limitations when it comes to modelling relational knowledge. For this reason, we finally discussed a number of settings in which vectors and symbols are combined in a looser way. Essentially, the underlying idea is to exploit the similarity structure captured by the vector space to identify symbolic knowledge that plausibly, but not deductively, follows from a given knowledge base.

References

1. Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. BoxE: A box embedding model for knowledge base completion. In *NeurIPS*, 2020.
2. Thomas Ager. *Disentangling low-dimensional vector space representations of text documents*. PhD thesis, Cardiff University, 2021.
3. Thomas Ager, Ondrej Kuzelka, and Steven Schockaert. Modelling salient features as directions in fine-tuned semantic spaces. In *CoNLL*, pages 530–540, 2018.
4. Rana Alshaikh, Zied Bouraoui, Shelan S. Jeawak, and Steven Schockaert. A mixture-of-experts model for learning multi-facet entity embeddings. In *COLING*, pages 5124–5135, 2020.
5. Rana Alshaikh, Zied Bouraoui, and Steven Schockaert. Learning conceptual spaces with disentangled facets. In *CoNLL*, pages 131–139, 2019.
6. Rana Alshaikh, Zied Bouraoui, and Steven Schockaert. Hierarchical linear disentanglement of data-driven conceptual spaces. In *IJCAI*, pages 3573–3579, 2020.
7. Abhijeet Awasthi, Sabyasachi Ghosh, Rasna Goyal, and Sunita Sarawagi. Learning from rules generalizing labeled exemplars. In *ICLR*, 2020.
8. Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In *IJCAI*, pages 364–369, 2005.
9. Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.
10. Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.
11. Nelly Barbot, Laurent Miclet, and Henri Prade. Analogy between concepts. *Artificial Intelligence*, 275:487–539, 2019.
12. Sabri Bayoudh, Laurent Miclet, and Arnaud Delhay. Learning by analogy: A classification rule for binary and nominal data. In *IJCAI*, pages 678–683, 2007.
13. Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond J. Mooney. Montague meets Markov: Deep semantics with probabilistic logical form. In **SEM*, pages 11–21, 2013.
14. Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.
15. Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz. Recent advances in querying probabilistic knowledge bases. In *IJCAI*, pages 5420–5426, 2018.
16. Zied Bouraoui, José Camacho-Collados, Luis Espinosa Anke, and Steven Schockaert. Modelling semantic categories using conceptual neighborhood. In *AAAI*, pages 7448–7455. AAAI Press, 2020.
17. Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. Inductive reasoning about ontologies using conceptual spaces. In *AAAI*, pages 4364–4370, 2017.
18. Zied Bouraoui and Steven Schockaert. Learning conceptual space representations of interrelated concepts. In Jérôme Lang, editor, *IJCAI*, pages 1760–1766, 2018.
19. Camille Bourgaux, Ana Ozaki, and Jeff Z. Pan. Geometric models for (temporally) attributed description logics. In Martin Homola, Vladislav Ryzhikov, and Renate A. Schmidt, editors, *DL*, 2021.

20 Katarina Britz, Thomas Meyer, and Ivan Varzinczak. Semantic foundation for preferential description logics. In *Australasian Joint Conference on Artificial Intelligence*, pages 491–500. Springer, 2011.

21 Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *J. Artif. Intell. Res.*, 48:115–174, 2013.

22 José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence*, 240:36–64, 2016.

23 Claudia d'Amato, Nicola Fanizzi, Bettina Fazzinga, Georg Gottlob, and Thomas Lukasiewicz. Ontology-based semantic search on the web and its combination with the power of inductive reasoning. *Ann. Math. Artif. Intell.*, 65(2-3):83–121, 2012.

24 Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. Lifted rule injection for relation embeddings. In *EMNLP*, pages 1389–1399, 2016.

25 Joaquín Derrac and Steven Schockaert. Inducing semantic relations from conceptual spaces: A data-driven approach to plausible reasoning. *Artif. Intell.*, 228:66–94, 2015.

26 Didier Dubois, Henri Prade, Francesc Esteva, Pere Garcia, and Lluis Godo. A logical approach to interpolation based on similarity relations. *International Journal of Approximate Reasoning*, 17(1):1–36, 1997.

27 Peter Gärdenfors. *Conceptual spaces: The geometry of thought.* MIT press, 2000.

28 Peter Gärdenfors. How to make the semantic web more semantic. In A.C. Varzi and L. Vieu, editors, *Formal Ontology in Information Systems*, pages 19–36. IOS Press, 2004.

29 Peter Gärdenfors. *The geometry of meaning: Semantics based on conceptual spaces.* MIT press, 2014.

30 Peter Gärdenfors and Mary-Anne Williams. Reasoning about categories in conceptual spaces. In *IJCAI*, pages 385–392, 2001.

31 Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170, 1983.

32 Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. Semantic characterization of rational closure: From propositional logic to description logics. *Artificial Intelligence*, 226:1–33, 2015.

33 Robert L Goldstone. Isolated and interrelated concepts. *Memory & Cognition*, 24(5):608–628, 1996.

34 Nelson Goodman. *Fact, fiction, and forecast.* Harvard University Press, 1955.

35 Georg Gottlob, Michael Morak, and Andreas Pieris. Recent advances in datalog ˆ\pm. In *Reasoning Web*, volume 9203 of *Lecture Notes in Computer Science*, pages 193–217, 2015.

36 Víctor Gutiérrez-Basulto, Jean Christoph Jung, Carsten Lutz, and Lutz Schröder. Probabilistic description logics for subjective uncertainty. *J. Artif. Intell. Res.*, 58:1–66, 2017.

37 Víctor Gutiérrez-Basulto and Steven Schockaert. From knowledge graph embedding to ontology embedding? an analysis of the compatibility between vector space representations and rules. In *KR*, pages 379–388, 2018.

38 Douglas R Hofstadter, Melanie Mitchell, et al. The copycat project: A model of mental fluidity and analogy-making. *Advances in Connectionist and Neural Computation Theory*, 2:205–267, 1995.

39 Keith J Holyoak and Paul Thagard. The analogical mind. *American psychologist*, 52(1):35–44, 1997.

40 Nicolas Hug, Henri Prade, Gilles Richard, and Mathieu Serrurier. Analogical classifiers: A theoretical perspective. In *ECAI*, pages 689–697, 2016.

41 Yazmín Ibáñez-García, Víctor Gutiérrez-Basulto, and Steven Schockaert. Plausible reasoning about el-ontologies using concept interpolation. In *KR*, pages 506–516, 2020.

42 Sarit Kraus, Daniel Lehmann, and Menachem Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial intelligence*, 44(1-2):167–207, 1990.

43 Tao Li and Vivek Srikumar. Augmenting neural networks with first-order logic. In *ACL*, pages 292–302, 2019.

44 Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. DeepProbLog: Neural probabilistic logic programming. In *NeurIPS*, pages 3753–3763, 2018.

45 Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *NAACL-HLT*, pages 746–751, 2013.

46 Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

47 Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, pages 809–816, 2011.

48 Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *NIPS*, 30:6338–6347, 2017.

49 Robert M Nosofsky. Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 10(1):104–114, 1984.

50 Daniel N Osherson, Edward E Smith, Ormond Wilkie, Alejandro Lopez, and Eldar Shafir. Category-based induction. *Psychological Review*, 97(2):185–200, 1990.

51 Matías Osta-Vélez and Peter Gärdenfors. Category-based induction in conceptual spaces. *Journal of Mathematical Psychology*, 96, 2020.

52 Özgür Lütfü Özçep, Mena Leemhuis, and Diedrich Wolter. Cone semantics for logics with negation. In *IJCAI*, pages 1820–1826, 2020.

53 Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

54 Henri Prade and Gilles Richard, editors. *Computational Approaches to Analogical Reasoning: Current Trends*, volume 548 of *Studies in Computational Intelligence*. Springer, 2014.

55 Henri Prade and Gilles Richard. From analogical proportion to logical proportions: A survey. In *Computational Approaches to Analogical Reasoning: Current Trends*, pages 217–244. 2014.

56 Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. In *ICLR*, 2020.

57 W.V. Quine. *From a Logical Point of View*. Harvard University Press, 1953.

58 Lance J Rips. Inductive judgments about natural categories. *Journal of Verbal Learning and Verbal Behavior*, 14(6):665–681, 1975.

59 Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. In *NIPS*, pages 3788–3800, 2017.

60 Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Matinata, and Paolo Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data*, 15(2):1–49, 2021.

61 Steven Schockaert, Yazmín Ibáñez-García, and Víctor Gutiérrez-Basulto. A description logic for analogical reasoning. In *IJCAI*, pages 2040–2046. ijcai.org, 2021.

62 Steven Schockaert, Yazmín Angélica Ibáñez-García, and Víctor Gutiérrez-Basulto. Modelling concept interpolation in description logics using abstract betweenness relations. In *DL*, 2021.

63 Steven Schockaert and Henri Prade. Solving conflicts in information merging by a flexible interpretation of atomic propositions. *Artif. Intell.*, 175(11):1815–1855, 2011.

64 Steven Schockaert and Henri Prade. Interpolative and extrapolative reasoning in propositional theories using qualitative knowledge about conceptual spaces. *Artif. Intell.*, 202:86–131, 2013.

65 Mikhail Sheremet, Dmitry Tishkovsky, Frank Wolter, and Michael Zakharyaschev. A logic for concepts and similarity. *Journal of Logic and Computation*, 17(3):415–452, 2007.

66 Steven A Sloman. Feature-based induction. *Cognitive Psychology*, 25(2):231–280, 1993.

67 Gustav Sourek, Vojtech Aschenbrenner, Filip Zelezný, Steven Schockaert, and Ondrej Kuzelka. Lifted relational neural networks: Efficient learning of latent relational structures. *J. Artif. Intell. Res.*, 62:69–100, 2018.

68 Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *ICLR*, 2018.

69 Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge graph embedding by relational rotation in complex space. In *ICLR*, 2019.

70 Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and Guillaume Bouchard. Knowledge graph completion via complex tensor factorization. *J. Mach. Learn. Res.*, 18:130:1–130:38, 2017.

71 Asahi Ushio, José Camacho-Collados, and Steven Schockaert. Distilling relation embeddings from pretrained language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *EMNLP*, pages 9044–9062, 2021.

72 Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

73 Rudolf Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In *Ordered Sets*, pages 445–470. 1982.

74 Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. In *ICML*, pages 5498–5507, 2018.

75 Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, 2015.

76 Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, 2015.

77 Zhun Yang, Adam Ishay, and Joohyung Lee. Neurasp: Embracing neural networks into answer set programming. In *IJCAI*, pages 1755–1762, 2020.

78 LA Zadeh. Calculus of fuzzy restrictions. In *Fuzzy Sets and Their Applications to Cognitive and Decision Processes: Proceedings of the US–Japan Seminar on Fuzzy Sets and Their Applications, Held at the University of California, Berkeley, California, July 1-4, 1974*, pages 1–39, 1975.

79 Zhanqiu Zhang, Jie Wang, Jiajun Chen, Shuiwang Ji, and Feng Wu. ConE: Cone embeddings for multi-hop reasoning over knowledge graphs. *NeurIPS*, 2021.