



HAL
open science

Integrating Ontologies and Vector Space Embeddings Using Conceptual Spaces (Invited Paper)

Zied Bouraoui, Víctor Gutiérrez-Basulto, Steven Schockaert

► **To cite this version:**

Zied Bouraoui, Víctor Gutiérrez-Basulto, Steven Schockaert. Integrating Ontologies and Vector Space Embeddings Using Conceptual Spaces (Invited Paper). Doctoral. Bergen, Norway. 2022. hal-04405120

HAL Id: hal-04405120

<https://hal.science/hal-04405120v1>

Submitted on 19 Jan 2024


HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.


1 Integrating Ontologies and Vector Space 2 Embeddings using Conceptual Spaces

3 Zied Bouraoui ✉

4 CRIL Laboratory, Université d'Artois, France

5 Víctor Gutiérrez-Basulto ✉ 

6 School of Computer Science & Informatics, Cardiff University, UK

7 Steven Schockaert ✉ 

8 School of Computer Science & Informatics, Cardiff University, UK

9 — Abstract —

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

25 **2012 ACM Subject Classification** Computing methodologies → Knowledge representation and
26 reasoning

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

29 **Digital Object Identifier** 10.4230/OASICS.AIB.2022.3

30 **Funding** This work has been supported by a grant from the Leverhulme Trust (project RPG-2021-
31 140).

32 **1** Introduction

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

36 ► **Example 1.** Consider the following rules:

37 $\text{expertInAI}(X) \leftarrow \text{authorOf}(X, Y), \text{hasTopic}(Y, \text{artificialIntelligence})$
38 $\text{hasTopic}(X, \text{artificialIntelligence}) \leftarrow \text{hasTopic}(X, \text{knowledgeRepresentation})$
39 $\text{hasTopic}(X, \text{artificialIntelligence}) \leftarrow \text{hasTopic}(X, \text{machineLearning})$
40 $\text{hasTopic}(X, \text{artificialIntelligence}) \leftarrow \text{hasTopic}(X, \text{multiAgentSystems})$
41 $\text{hasTopic}(X, \text{artificialIntelligence}) \leftarrow \text{hasTopic}(X, \text{naturalLanguageProcessing})$
42



© Zied Bouraoui, Víctor Gutiérrez-Basulto and Steven Schockaert;
licensed under Creative Commons License CC-BY 4.0

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



Open Access Series in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

3:2 Integrating Ontologies and Vector Space Embeddings using Conceptual Spaces

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

49 $\{\text{authorOf}(\text{bob}, p), \text{hasTopic}(p, \text{knowledgeRepresentation})\}$
50

51 Given this set of facts, together with the aforementioned rules, we can conclude that
52 $\text{hasTopic}(p, \text{artificialIntelligence})$ holds and thus also that $\text{expertInAI}(\text{bob})$ holds.

53 Using ontologies for encoding conceptual knowledge has at least two key advantages. First,
54 the formal semantics of the underlying logic ensures that knowledge can be encoded in a
55 precise and unambiguous way. This, in turn, ensures that different applications can rely on a
56 shared understanding of the meaning of the concepts involved. Second, ontologies enable
57 us to capture knowledge in a transparent and interpretable way¹, which makes it relatively
58 straightforward to update knowledge and to support decisions with meaningful explanations.
59 But ontologies, and symbolic approaches to knowledge representation more generally, also
60 have important drawbacks. A first issue stems from the fact that the knowledge which is
61 captured in an ontology is rarely complete. For instance, consider the following set of facts:

62 $\{\text{authorOf}(\text{alice}, q), \text{hasTopic}(q, \text{planning})\}$
63

64 As none of the available rules express that planning is a sub-field of AI, we cannot infer that
65 $\text{expertInAI}(\text{alice})$ holds. Nonetheless, to a human observer, it seems clear that this would
66 be a valid inference, even without a precise understanding of what the predicate expertInAI
67 is supposed to capture. Essentially, standard frameworks for modelling ontologies lack a
68 mechanism for inductive reasoning [28]. This is not something which can be easily addressed,
69 as inductive arguments rely on graded notions such as similarity and typicality [58, 50, 66, 51].
70 Another issue is that many concepts are difficult to characterise in a satisfactory way using
71 logical rules. For instance, somebody with a single published paper in AI would not normally
72 be considered to be an AI expert, except perhaps if the work was particularly influential
73 or groundbreaking, but formalising such notions using rules is challenging. Probabilistic
74 extensions of standard ontology languages [36, 15] may alleviate some of the aforementioned
75 issues, but such frameworks still do not allow us to model similarity, or aspects that are a
76 matter of degree (e.g. being an expert in AI).

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

83 $K = \{(\text{bob}, \text{authorOf}, p), (p, \text{hasTopic}, \text{knowledgeRepresentation}),$
84 $(p, \text{hasTopic}, \text{artificialIntelligence}), (\text{bob}, \text{hasProperty}, \text{expertInAI})\}$
85

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.

86 Approaches for Knowledge graph embedding (KGE) learn a vector representation $\mathbf{e} \in \mathbb{R}^n$ for
 87 each entity e and a scoring function $\phi_r : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ for each relation type r , such that
 88 $\phi_r(\mathbf{e}, \mathbf{f})$ captures the plausibility of the triple (e, r, f) , i.e. the plausibility that the relation r
 89 holds between the entities e and f [14, 75, 70, 69]. The vector \mathbf{e} is called the *embedding* of
 90 entity e . The purpose of KGE is at least two-fold. First, it is hoped that this embedding
 91 will uncover some of the underlying semantic dependencies in the KG, and that as a result,
 92 we will be able to identify plausible triples that are missing from the given KG. Second, by
 93 encoding the information that is captured in the knowledge graph using vectors, it becomes
 94 easier to exploit this information in neural network models.

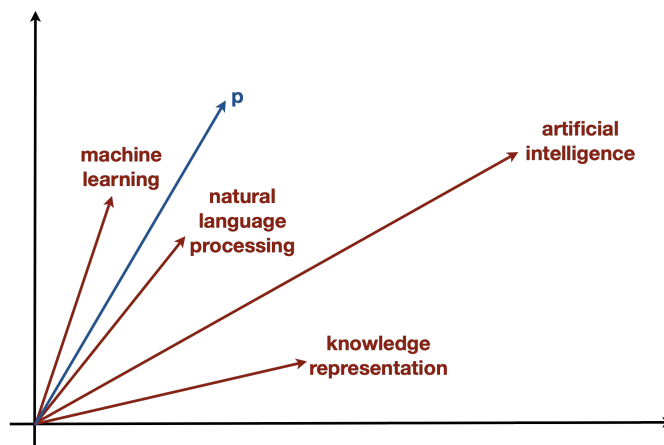
Figure 1 shows a vector encoding of the paper p and some of the considered subject areas. For this example, we assume that the dot product between p and a subject area indicates how relevant that subject area is to p , i.e. we have $\phi_{\text{hasTopic}}(\mathbf{e}, \mathbf{f}) = \mathbf{e} \cdot \mathbf{f}$. 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 $\mathbf{p} \cdot \mathbf{v}_{\text{ML}} \approx \mathbf{p} \cdot \mathbf{v}_{\text{NLP}} > \mathbf{p} \cdot \mathbf{v}_{\text{KR}}$, 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:

$$\mathbf{v}_{\text{ML}} \cdot \mathbf{x} < \mathbf{v}_{\text{AI}} \cdot \mathbf{x}$$

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

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

119 The first question we address in this paper is whether a tighter integration of rules and
 120 vector representations is possible. The main idea is to view symbolic knowledge as qualitative
 121 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 .

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

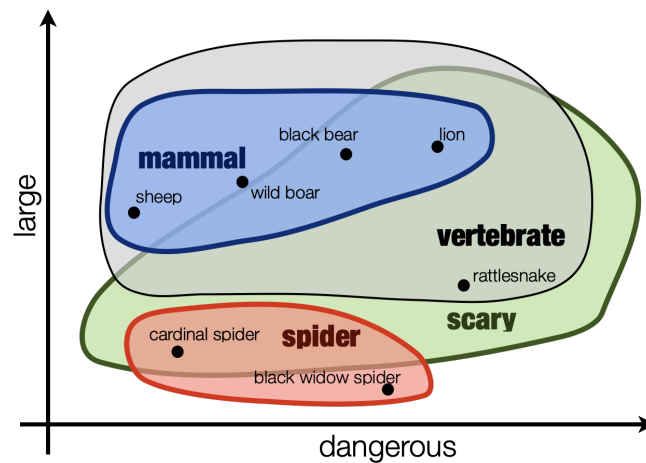
- 128 ■ In practice, it is often difficult to learn region-based representations of concepts from
 129 data.
- 130 ■ Conceptual space representations cannot be used for modelling relational knowledge, e.g.
 131 rules involving binary predicates.

132 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:

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

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



■ **Figure 2** Illustration of a conceptual space of animals.

2 Background

135

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

2.1 Conceptual Spaces

141

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

155 While it is convenient to think about conceptual spaces as vector space embeddings with
 156 some added structure, conceptual spaces do not necessarily have the structure of a vector
 157 space. A conceptual space is defined from a set of *quality dimensions* Q_1, \dots, Q_n . Each of
 158 these quality dimensions captures a primitive feature. As a standard example, the conceptual
 159 space of colours is built from three quality dimensions, representing hue, saturation and
 160 intensity. A distinction is made between *integral* and *separable* quality dimensions. Intuitively,
 161 separable quality dimensions are those that have a meaning on their own. For instance, *size*
 162 could be seen as a separable dimension. On the other hand, *hue* is not separable, as we
 163 cannot imagine the hue of a colour without also specifying its saturation and intensity. This

164 distinction between integral and separable dimensions plays an important role in cognitive
 165 theories, as it affects how similarity is perceived. For instance, Euclidean distance is normally
 166 used when integral dimensions need to be combined, whereas Manhattan distance is used
 167 when separable dimensions need to be combined [49, 27]. Quality dimensions are partitioned
 168 into so-called *domains*, where dimensions that belong to the same domain are assumed to be
 169 integral, while dimensions from different domains are assumed to be separable. For instance,
 170 a conceptual space of physical objects may be composed of three domains: the colour domain
 171 (containing the hue, saturation and intensity quality dimensions), the size domain (containing
 172 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, \dots, Q_k then the elements of D_i are tuples $(x_1, \dots, x_k) \in Q_1 \times \dots \times Q_k$. 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, \dots, 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 $\text{bet}_i \subseteq D_i \times D_i \times D_i$. A region $R \subseteq D_i$ is then said to be *convex* iff

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

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

178 2.2 Ontology Languages

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

188 Existential Rules

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

$$193 \quad \exists X_1, \dots, X_j . H_1 \wedge \dots \wedge H_k \leftarrow B_1 \wedge \dots \wedge B_n, \quad (1)$$

194 where $n \geq 0$, $k \geq 1$, B_1, \dots, B_n and H_1, \dots, H_k are atoms with terms in $\mathbf{C} \cup \mathbf{V}$, and
 195 $X_1, \dots, X_j \in \mathbf{V}$. From here on, we assume w.l.o.g. that $k = 1$ [21] and we omit the subscript
 196 in H_1 . We further allow *negative constraints* (also simply called *constraints*), which are
 197 expressions of the form $\perp \leftarrow B_1 \wedge \dots \wedge B_n$, where the B_i s are as above and \perp denotes the
 198 truth constant *false*. A finite set Σ of existential rules and constraints is called an *ontology*.
 199 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
 200 terms in \mathbf{C} . A *knowledge base (KB)* \mathcal{K} is a pair (Σ, D) with Σ an ontology and D a database.

201 **Semantics** An *interpretation* \mathcal{I} over \mathfrak{R} is a (possibly infinite) set of atoms over \mathfrak{R} with
 202 terms in $\mathbf{C} \cup \mathbf{N}$. An interpretation \mathcal{I} is a *model* of Σ if it satisfies all rules and constraints:
 203 $\{B_1, \dots, B_n\} \subseteq \mathcal{I}$ implies $\{H\} \subseteq \mathcal{I}$ for every existential rule σ in Σ , where existential
 204 variables can be witnessed by constants or labelled nulls, and $\{B_1, \dots, B_n\} \not\subseteq \mathcal{I}$ for all
 205 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
 206 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
 207 if it has a model. We refer to elements in $\mathbf{C} \cup \mathbf{N}$ simply as *objects*, call atoms α containing
 208 only objects as terms *ground*, and denote with $\mathfrak{O}(\mathcal{I})$ the set of all objects occurring in \mathcal{I} .

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

$$211 \quad \text{husband}(Y) \leftarrow \text{wife}(X) \wedge \text{married}(X, Y) \quad (2)$$

$$212 \quad \exists X. \text{husband}(X) \wedge \text{married}(X, Y) \leftarrow \text{wife}(Y) \quad (3)$$

$$213 \quad \perp \leftarrow \text{husband}(X) \wedge \text{wife}(X) \quad (4)$$

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

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

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

217 \mathcal{EL} -family

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

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

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

221 For instance, $A \sqcap (\exists r.(B \sqcap C))$ is an example of a well-formed \mathcal{EL}_\perp concept, assuming
 222 $A, B, C \in \mathbf{N}_\mathbf{C}$ and $r \in \mathbf{N}_\mathbf{R}$. The fragment of \mathcal{EL}_\perp in which \perp is not used is known as \mathcal{EL} . An
 223 \mathcal{EL}_\perp *TBox (ontology)* \mathcal{T} is a finite set of *concept inclusions (CIs)* of the form $C \sqsubseteq D$, where
 224 C, D are \mathcal{EL}_\perp concepts.

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

$$227 \quad \exists \text{married.Wife} \sqsubseteq \text{Husband} \tag{5}$$

$$228 \quad \text{Wife} \sqsubseteq \exists \text{married.Husband} \tag{6}$$

$$229 \quad \text{Husband} \sqcap \text{Wife} \sqsubseteq \perp \tag{7}$$

231 **Semantics** The semantics of description logics are usually given in terms of first-order
 232 interpretations $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. Such interpretations consist of a nonempty *domain* $\Delta^{\mathcal{I}}$ and an
 233 *interpretation function* $\cdot^{\mathcal{I}}$, which maps each concept name A to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each
 234 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
 235 to complex concepts as follows:

$$236 \quad (\top)^{\mathcal{I}} = \Delta^{\mathcal{I}}, \quad (\perp)^{\mathcal{I}} = \emptyset \quad (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}},$$

$$237 \quad (\exists r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \exists d' \in C^{\mathcal{I}}, (d, d') \in r^{\mathcal{I}}\}.$$

239 We now introduce two classical reasoning tasks. An interpretation \mathcal{I} *satisfies* a concept
 240 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
 241 \mathcal{T} if it satisfies all CIs in \mathcal{T} . A concept C *subsumes a concept* D *relative to a TBox* \mathcal{T} if
 242 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
 243 *satisfiable w.r.t.* \mathcal{T} if there is a common model of C and \mathcal{T} .

244 2.3 Knowledge Graph Embedding

245 Let a set of entities \mathcal{E} and a set of binary relations \mathcal{R} be given. A knowledge graph (KG)
 246 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
 247 (e, r, f) . These triples encode the fact that the relation r holds between the entities e and
 248 f . For instance, we may have a triple such as $(\text{london}, \text{capitalOf}, \text{uk})$, encoding that London
 249 is the capital of the UK. A knowledge graph is thus essentially a set of relational facts,
 250 with the limitation that all relations are binary. Note, however, that the set of entities \mathcal{E}
 251 typically includes both individuals (i.e. constants referring to specific objects, e.g. london)
 252 and attribute values, which allow us to encode unary predicates. For instance, the relational
 253 fact $\text{scary}(\text{lion})$ Could be encoded as the KG triple $(\text{lion}, \text{hasAttribute}, \text{scary})$.

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 $\phi_r : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ for each $r \in \mathcal{R}$. 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:

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

254 Different models differ in which constraints they put on the matrix \mathbf{M}_r . For instance, in
255 the RESCAL model [47] this matrix is unconstrained, whereas DistMult [76] only allows
256 diagonal matrices. In recent years, several authors have focused on designing models that
257 make it easier to capture certain relational structures. For instance, embeddings based
258 on hyperbolic geometry have been used to make it easier to model hierarchical structures,
259 such as *is-a* and *part-of* hierarchies [48]. Region-based models, e.g. representing entities as
260 boxes or cones, have been used for their ability to model both hierarchies and intersections
261 [1, 52, 79]. In [68] a model is proposed in which relations are viewed as rotations, to facilitate
262 modelling relational composition, as well as properties such as symmetry. It should be noted,
263 however, that while these models can capture certain relational dependencies to some extent,
264 in most models there is no explicit link between a given knowledge graph embedding and the
265 relational dependencies it captures. Moreover, relatively little is known about which kinds
266 of dependencies different models are capable of capturing (or, more generally, which sets of
267 dependencies can be jointly captured). Of course, this first requires us to formalise what it
268 means for an embedding to capture a relational dependency. We will return to this question
269 in Section 4.

270 **3 Learning Conceptual Space Representations**

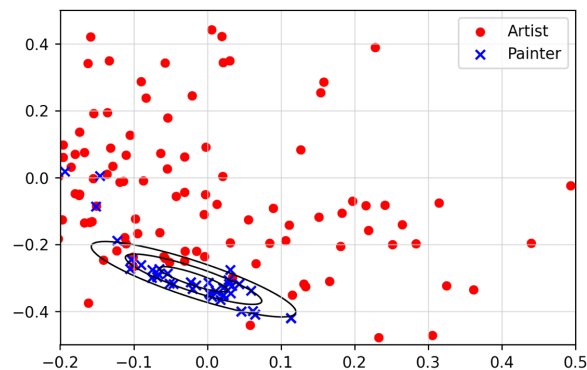
271 If we want to use conceptual spaces as an interface between symbolic ontologies and vectors
272 space embeddings, a crucial question is whether it is possible to learn conceptual spaces from
273 data. What matters in this context is (i) whether we can learn region-based representations
274 of concepts and (ii) whether we can learn vector representations in which dimensions are
275 meaningful and organised into domains. These two issues are discussed in Sections 3.1 and
276 3.2 respectively.

277 **3.1 Modelling Concepts as Regions**

278 **Learning Gaussian Representations** In learned vector space embeddings, the objects from
279 some domain of interest are represented as points or vectors, as in conceptual spaces. Most
280 embedding models do not learn region-based representations of concepts. However, if we
281 have access to a number of instances c_1, \dots, c_n of a given concept C , we can aim to learn
282 a region-based representation of C from embeddings of these instances. The potential of
283 this strategy stems from the fact that in many embedding models, these instances can
284 be expected to appear clustered together in the vector space. To illustrate this, consider
285 Figure 3, which shows the first two principal components of a 300-dimensional embedding of
286 BabelNet concepts [46] using NASARI vectors², which have been learned from Wikipedia
287 and are linked to BabelNet [22]. In the figure, the red points correspond to entities that are
288 instances of the concept *Artist*, while the blue points correspond to entities that are instances
289 of *Painter*. For instance, the embeddings of *Edouard Manet*, *Vanessa Bell* and *Claude Monet*
290 appear close to the centre of the blue point cloud. As can be seen, painters appear as a
291 distinct cluster in this vector space embedding.

292 When attempting to learn a region-based concept representation, we are faced with two
293 challenges: (i) we typically only have access to positive examples and (ii) the number of
294 available instances is often much smaller than the number of dimensions in the vector space.
295 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.

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

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

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

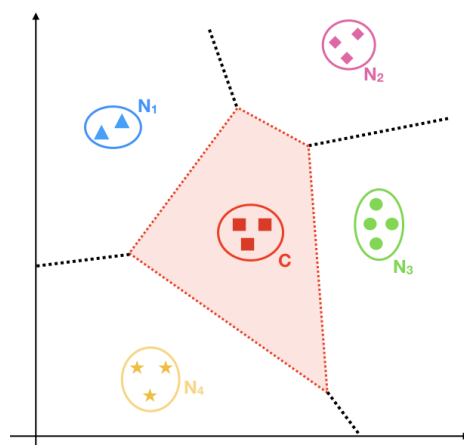
326 representation of concept A provides us with a prior on the representation of concept B , but
327 the representation of concept B also provides us with a prior on the representation of A .
328 This can be addressed using Gibbs sampling; we refer to [18] for the details.

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

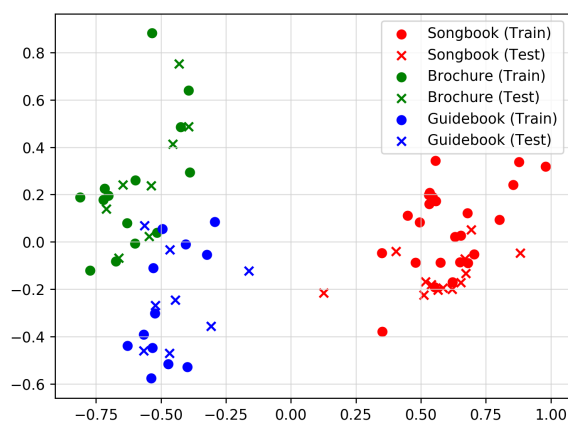
338 *Priors on Variance.* To obtain a prior on the variance of the Gaussian representing C , we
339 take the view that this variance should be similar to that of the concepts that are most
340 similar to C . To find such concepts, we could take the siblings of C in an ontology, the
341 concepts whose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

344 **Exploiting contrast sets** A common strategy for learning conceptual space representations
345 is to associate each concept with a single point, which intuitively represents its prototype
346 [30]. The region representing a given concept C then consists of all points that are closer
347 to the prototype of C than to the prototype of any other concept, i.e. concept regions are
348 obtained as the Voronoi tessellation of a set of prototype points. This strategy is appealing,
349 because it allows us to learn concept regions with a much wider extension than when learning
350 Gaussians, especially in cases where we only have a few instances per concept. The main
351 idea is illustrated in Figure 4, where we are interested in learning a region for the concept C .
352 When using Gaussians, we would end up with ellipsoidal regions (contours) similar to the
353 ones displayed in the figure. As a result, most points of the space are not assigned to any of
354 the concepts. In contrast, if we construct prototypes by averaging the embeddings of the
355 instances of a concept, and compute the resulting Voronoi tessellation, we essentially carve
356 up the space, as also illustrated in the figure. To see why this can be beneficial in practice,
357 Figure 5 shows the vector representations of the instances of three concepts: **Songbook**,
358 **Brochure** and **Guidebook**. Now consider the left-most test instance of **Songbook**. If we are
359 only given the training instances of this concept, this test instance is unlikely to be covered
360 by the resulting representation. In contrast, if we instead attempt to carve up the space into
361 regions corresponding to **Songbook**, **Brochure** and **Guidebook**, then this test instance would
362 be classified correctly. The problem with implementing the aforementioned idea is that it
363 only works if we are given a set of concepts that form a *contrast set* [33], i.e. a set of mutually
364 exclusive natural categories that exhaustively cover some sub-domain. For example, the set of
365 all common color names, the set **{Fruit, Vegetable}** and the set **{NLP, IR, ML}** can intuitively
366 be thought of as contrast sets. We say that two concepts are conceptual neighbours if they
367 belong to the same contrast set and compete for coverage (i.e. are adjacent in the resulting
368 Voronoi tessellation).

369 Existing ontologies do not typically describe contrast sets or conceptual neighbourhood.
370 To deal with this, [16] introduced a strategy for learning conceptual neighbourhood from
371 data, i.e. for discovering pairs of concepts that are conceptual neighbours. Note that they
372 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.

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

- 385 1. Learn a Gaussian representation of A and B from their given instances.
- 386 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.

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

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

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

399 The sentence suggests that *hamlet* and *village* are conceptual neighbors as it makes clear
400 that these concepts are closely related but different. Once a classifier is trained, based on
401 the weakly supervised training set, we can then apply it to other concepts. To learn the
402 representation of a given target concept C (e.g. a concept with only few known instances), we
403 can then use the text classifier to identify which of its siblings, in a given taxonomy, are most
404 likely to be conceptual neighbours, and determine the representation of C accordingly. Tables
405 2 and 3 show some examples of the top conceptual neighbor predicted by the text classifier,
406 for different target concepts. In particular, Table 3 shows examples where the resulting
407 concept representations (i.e. the representations of the target concepts obtained by exploiting
408 the predicted conceptual neighbourhood) were of high quality, as measured in terms of F1
409 score for held-out entities. Similarly, Table 2 shows examples where the resulting concept
410 representations were of low quality. As can be seen, the predicted conceptual neighbours
411 in Table 3 are clearly more meaningful than the predicted neighbours in Table 2. This
412 illustrates how the quality of the concept representations is closely linked to our ability to
413 find meaningful conceptual neighbours. Overall, the experiments in [16] showed that using
414 predicted conceptual neighbourhood, on average, led to much better concept representations
415 than when estimating Gaussians from the known instances of the target concept.

416 3.2 Learning Quality Dimensions

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

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

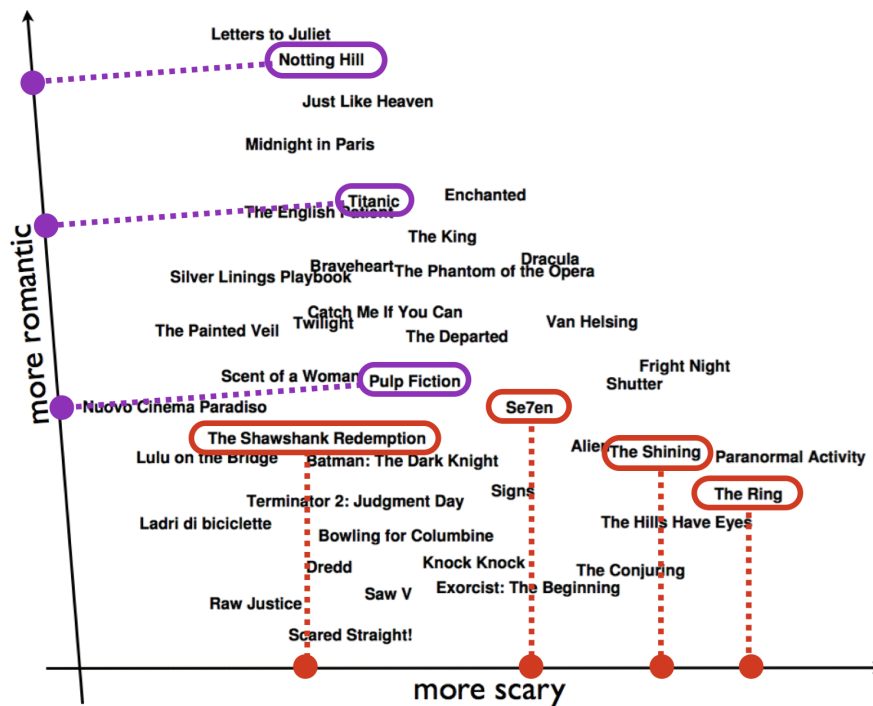
Concept	Top neighbor	F1
Amphitheater	Velodrome	67
Proxy server	Application server	61
Ketch	Cutter	74
Quintet	Brass band	67
Sand dune	Drumlin	71

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

419 the interpretability of learned vector space representations. In this section, we review work
 420 that has focused on mitigating this issue, by identifying interpretable directions in learned
 421 vector spaces. These interpretable directions can then play the role of quality dimensions.
 422 This is illustrated in Figure 6, which shows a two-dimensional projection of an embedding of
 423 movies from [25]. Along with the embedding of the movies themselves, the figure also shows
 424 two directions that have been identified: one direction which ranks the movies from least
 425 to most *scary*, and another direction which ranks the movies from least to most *romantic*.
 426 Formally, we say that the direction of some vector \mathbf{v} models a property P , such as *scary*, if
 427 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
 428 property P to a greater extent than entity e_2 .

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

448 ■ touching, inspirational, warmth, dignity, sadness, heartwarming, ...



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

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

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

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

466 While these directions express semantically meaningful properties, it would be more
 467 natural to represent such properties as regions than as quality dimensions. The fact that such
 468 thematic properties cannot be distinguished from the semantic attributes mentioned above
 469 is clearly a limitation of the method from [25]. In [2], it was found that the nature of the
 470 clusters, i.e. whether they intuitively correspond to quality dimensions rather than thematic
 471 properties, to some extent depends on the scoring function that is used for evaluating the

472 linear classifiers. However, regardless of the scoring function that is used, a mixture of
 473 different types of properties is found. One possible solution could be to require that clusters
 474 which correspond to quality dimensions should contain a sufficient proportion of adjectives,
 475 as clusters consisting mostly of nouns are more likely to be thematic properties. On the other
 476 hand, it is not clear that having thematic “quality dimensions” is necessarily problematic.
 477 While it makes the resulting representation different from a conceptual space, it still allows
 478 us to “disentangle” the vector representation into different aspects (e.g. genre, sentiment,
 479 emotion). Furthermore, a cluster of terms related to horror movies could still be viewed as a
 480 quality dimension if we view it as ranking movies based on how “horror-like” they are.

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

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

506 Note that we cannot learn these facets by simply clustering the quality dimensions. For
 507 instance, *thriller* and *scary* may be represented by similar directions in the vector space,
 508 but they should be assigned to different facets. In contrast, *romance* and *horror* would
 509 be represented by dissimilar directions but nonetheless belong to the same facet. The key
 510 solution, which was developed in [5] and [4], is to rely on word embeddings to identify words
 511 that describe properties of the same kind. For instance, the word embeddings of different
 512 movie genres tend to be similar, because such words tend to appear in similar contexts. In
 513 the same way, different adjectives describing emotions tend to be represented using similar
 514 word vectors. This suggests a simple strategy for learning facets: (i) cluster the word vectors
 515 of the words associated with the quality dimensions that were identified in the given vector
 516 space; and (ii) represent the facet by the vector space that is spanned by quality dimensions
 517 that are assigned to it. Unfortunately, this strategy was found to perform poorly in [5]. The
 518 main reason is that in many areas there is one dominant facet, such as the genre in the case

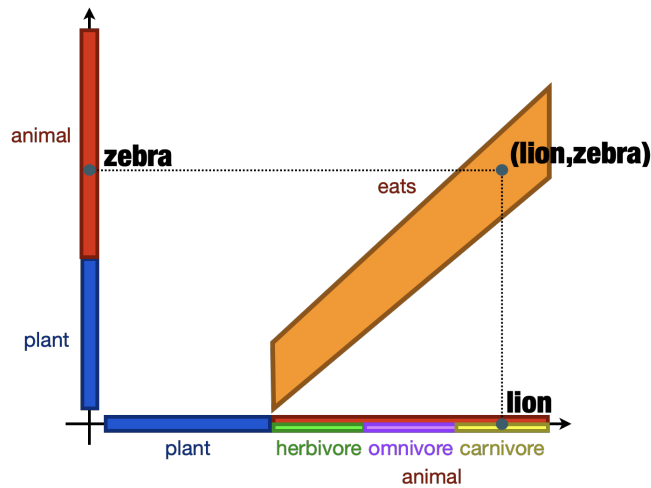
519 of movies. When applying the aforementioned strategy, what happens is that each of the
520 resulting facet-specific vector spaces mostly models the dominant facet. To address this issue,
521 [5] proposed an iterative strategy, in which the dominant facet is first identified and then
522 explicitly disregarded when determining the second facet, etc. Another practical challenge
523 is that the overall method is computationally demanding, especially the fact that a linear
524 classifier has to be learned for each word from the vocabulary, to identify the interpretable
525 directions (in the overall space and in each of the lower-dimensional facet-specific spaces). To
526 address this issue, [4] introduced a model that directly learns facet-specific vector spaces from
527 bag-of-words representations of the entities, using a mixture-of-experts model to generalize
528 the GloVe [53] word embedding model. Using this approach, facet-specific vector spaces can
529 be learned much more efficiently, and moreover the resulting embeddings tend to be of a
530 higher quality. The main limitation, however, is that this model assumes that suitable vector
531 spaces can be learned from bag-of-words representations (rather than being agnostic to how
532 the initial vector space embedding is learned) and that GloVe is a suitable embedding model
533 for learning these vector spaces.

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

545 **4** Modelling Relations with Conceptual Spaces

546 Conceptual spaces act as an interface between vector space embeddings and symbolic
547 knowledge. However, because conceptual spaces do not capture relational knowledge, they
548 are essentially limited to capturing Horn rules with unary predicates. In this section, we
549 explore whether the framework of conceptual spaces can be generalised to encode rules with
550 binary and higher arity relations. We focus on the analysis presented in [37] but use a
551 construction that is somewhat more intuitive than the one used in the latter paper. The
552 main idea is to view a k -ary relation as a convex region in the Cartesian product of k
553 conceptual spaces. For simplicity, in this section we will assume that conceptual spaces
554 correspond to Euclidean spaces. Each individual a is then represented as a vector $\mathbf{a} \in \mathbb{R}^n$. A
555 tuple (a_1, \dots, a_k) is represented as the concatenation of the vectors representing a_1, \dots, a_k , i.e.
556 (a_1, \dots, a_k) is represented as the $n \cdot k$ -dimensional vector $\mathbf{a}_1 \oplus \dots \oplus \mathbf{a}_k$, where we write \oplus for
557 vector concatenation.

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



■ **Figure 7** Illustration of a relational conceptual space.

564 representation captures the following rule:

$$565 \quad \text{carnivore}(X) \leftarrow \text{eats}(X, Y), \text{animal}(Y)$$

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

$$574 \quad \exists Y. \text{eats}(X, Y) \wedge \text{animal}(Y) \leftarrow \text{carnivore}(X)$$

576 To see why this rule is satisfied for the configuration depicted in Figure 7, consider a value
 577 $x \in \mathbb{R}$ which lies in the interval representing carnivore. Then we can always find a coordinate
 578 $y \in \mathbb{R}$ such that the point $\mathbf{p} = (x, y)$ lies in the region for eats and such that y lies in the
 579 interval modelling animal. In Section 4.1 we discuss these intuitions in more detail. We also
 580 provide a characterisation about the kinds of relational rules that can be modelled using
 581 convex regions. Subsequently, in Section 4.2 we discuss the relationship with knowledge
 582 graph embedding models.

583 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 $r(a_1, \dots, a_k)$ is satisfied if the representation of the tuple (a_1, \dots, a_k) lies in the region representing r , i.e.:

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

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

nRI

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:

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

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

$$586 \quad r(X_1, \dots, X_k) \leftarrow s(X_1, \dots, X_k), t(X_1, \dots, X_k) \quad (8)$$

The corresponding geometric constraint is as follows:

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

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

$$591 \quad r(X, Z) \leftarrow s(X, Y), t(Y, Z) \quad (9)$$

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 $I = \{i_1, \dots, i_l\} \subseteq \{1, \dots, k\}$ be a set of indices. For a point $(x_1, \dots, x_{k \cdot n}) \in \mathbb{R}^{k \cdot n}$, we define its *restriction to I* as follows

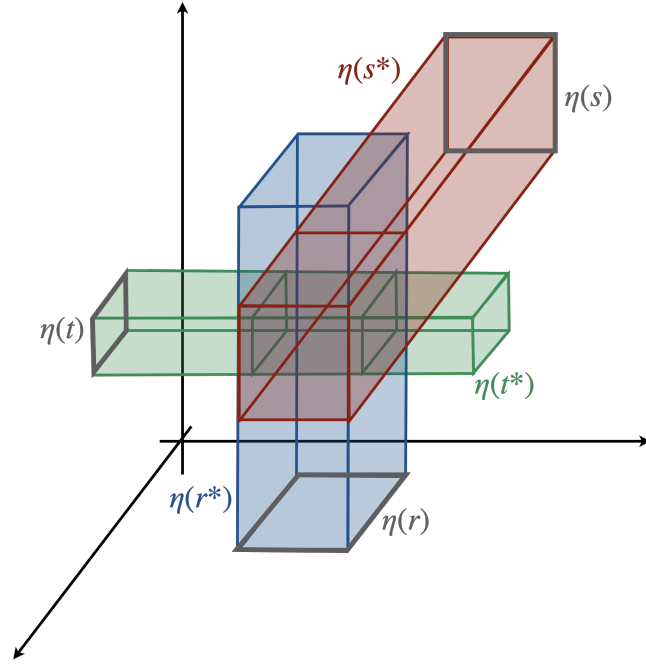
$$(x_1, \dots, x_{k \cdot n}) \downarrow I = \bigoplus_{i \in I} (x_{n \cdot i + 1}, \dots, x_{n \cdot i + n})$$

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

$$598 \quad \text{ext}_I^k(R) = \{\mathbf{x} \in \mathbb{R}^{k \cdot n} \mid \mathbf{x} \downarrow I \in R\}$$

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

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



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

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

$$604 \quad \exists Y . r(X, Y) \wedge s(Y, Z) \leftarrow t(X, Z) \quad (10)$$

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 $\text{ext}_{\{1,2\}}^3(\eta(r)) \cap \text{ext}_{\{2,3\}}^3(\eta(s))$. 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:

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

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

A central question is which kinds of rules can be faithfully⁴ 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 $\perp \leftarrow r_1(X, Y), r_2(X, Y)$, 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:

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

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

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

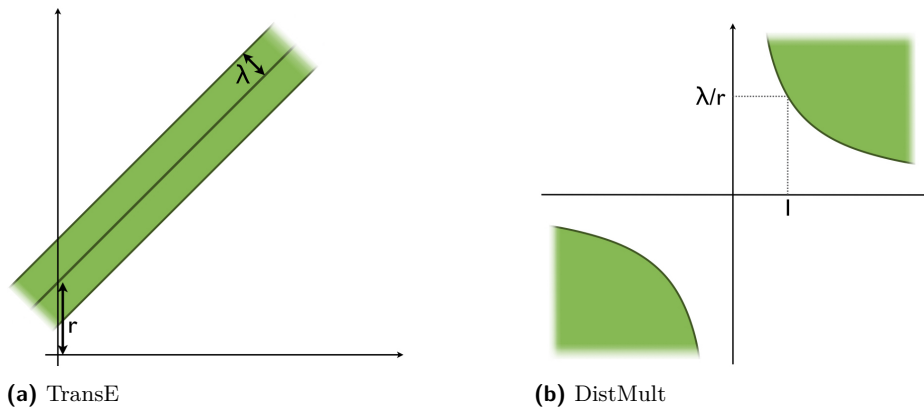
620 Finally, it should be noted that the restriction to arbitrary convex regions means that
 621 negation and disjunction cannot easily be modelled. Some authors have proposed geometric
 622 models that were specifically designed with such logical connectives in mind, including the
 623 use of axis aligned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

625 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:

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

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



■ **Figure 9** Region based view of knowledge graph embedding models.

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

630 **5** Plausible Symbolic Reasoning using Vector Space Embeddings

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

640 One of the most natural solutions is to use vector representations to implement a form
 641 of similarity based reasoning [23, 13]. For instance, we could have a KB with factual
 642 knowledge stating that strawberries are instances of the concept berries, $\text{Berry}(\text{strawberry})$,
 643 and ontological knowledge stating that berries are healthy, $\text{Berry} \sqsubseteq \text{Healthy}$. Clearly, this
 644 KB entails that strawberries are healthy. Further, using a standard word embedding [45],
 645 we can find out that `strawberry` and `raspberry` are highly similar. Now, using the KB and
 646 the additional similarity information, we can infer that it is plausible that raspberries are
 647 berries and, therefore, healthy. This same idea could be lifted to find the similarity between
 648 concept names (classes) and find plausible rules. For instance, assume that strawberries
 649 and raspberries are concept names and that our ontology specifies that strawberries are
 650 healthy, i.e. $\text{Strawberry} \sqsubseteq \text{Healthy}$. Using the similarity between strawberries and raspberries,
 651 we could then infer that the concept inclusion $\text{Raspberry} \sqsubseteq \text{Healthy}$ is plausible. However,
 652 implementing this strategy in a principled way is difficult, because it is unclear how we
 653 can formally relate degrees of similarity to the plausibility of the inferred rules, i.e. if we
 654 can infer using standard deduction that $C_1 \sqsubseteq X$, how similar does concept C_2 needs to be
 655 to C_1 to accept the plausible inference $C_2 \sqsubseteq X$? For this reason, rather than focusing on
 656 similarity based reasoning, it has been proposed to focus on *interpolative reasoning* instead
 657 [64]. The main difference is that instead of focusing on the similarity between two entities, we
 658 focus on how one entity relates to a group of entities. For instance, we say that the concept

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

670 5.1 Interpolative Reasoning

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

$$673 \quad \text{Strawberry} \sqsubseteq C \quad \text{Orange} \sqsubseteq C$$

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

$$677 \quad \text{Raspberry} \sqsubseteq C \tag{11}$$

679 This conclusion relies on background knowledge about strawberries, oranges and raspberries,
 680 in particular the fact that raspberries are expected to have all the *natural* properties that
 681 strawberries and oranges have in common (e.g. being high in vitamin C). In such a case, we say
 682 that raspberries are *conceptually between* strawberries and oranges. Importantly, knowledge
 683 about conceptual betweenness can be derived from data-driven representations. For instance,
 684 [25] found that geometric betweenness closely corresponds to conceptual betweenness in
 685 vector spaces learned with multi-dimensional scaling.

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

696 The extension \mathcal{EL}^{\boxtimes} of \mathcal{EL} was designed based on the above intuitions, with the aim of
 697 enabling reasoning about conceptual betweenness and natural concepts, and thus supporting
 698 interpolative reasoning. Syntactically, \mathcal{EL} is extended with the in-between constructor, which
 699 allows us to describe the set of objects that are between two concepts: we write $C \boxtimes D$ to
 700 denote all objects that are between the concepts C and D . We further assume that \mathbb{N}_C
 701 contains a distinguished infinite set of *natural concept names* $\mathbb{N}_C^{\text{Nat}}$. The syntax of \mathcal{EL}^{\boxtimes}
 702 *concepts* C, D is thus defined by the following grammar, where $A \in \mathbb{N}_C$, $A' \in \mathbb{N}_C^{\text{Nat}}$ and
 703 $r \in \mathbb{N}_R$:

$$704 \quad C, D := \top \mid A \mid C \sqcap D \mid \exists r.C \mid N \quad N, N' := A' \mid N \sqcap N' \mid N \boxtimes N'$$

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

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

$$709 \quad \text{Strawberry} \sqsubseteq \text{Healthy} \tag{12}$$

$$710 \quad \text{Orange} \sqsubseteq \text{Healthy} \tag{13}$$

$$711 \quad \text{Raspberry} \sqsubseteq \text{Strawberry} \bowtie \text{Orange} \tag{14}$$

$$712 \quad \text{Healthy} \sqsubseteq \exists \text{improves. QualityOfLife} \tag{15}$$

714 such that $\text{Strawberry}, \text{Orange}, \text{Raspberry}, \text{Healthy} \in \mathbb{N}_C^{\text{Nat}}$.

715 The semantics of \mathcal{EL}^{\bowtie} needs to adequately characterise natural concepts and concept
716 betweenness, and thus support interpolation, i.e.: such that from $A \sqsubseteq B_1 \bowtie B_2$, $B_1 \sqsubseteq C$ and
717 $B_2 \sqsubseteq C$, we can derive $A \sqsubseteq C$, provided that C is *natural*. To this end, Ibáñez-García et
718 al. [41] proposed two semantics: a feature-enriched semantics inspired by formal concept
719 analysis [73] and a geometric semantics inspired by conceptual spaces. In the former, at the
720 semantic level a set of features is associated with each concept. Note that these features are
721 semantic constructs, which have no direct counterpart at the syntactic level. A concept is then
722 natural if it is completely characterized by these features, while B is between A and C if the
723 set of features associated with B contains the intersection of the sets associated with A and C .
724 In the second semantics, concepts are interpreted as regions from a vector space. A concept is
725 then natural if it is interpreted as a convex region, while B is between A and C if the region
726 corresponding to B is geometrically between the regions corresponding to A and C (i.e. in the
727 convex hull of their union). We refrain from giving the full technical details, but invite the
728 interested reader to look at [41]. Ibáñez-García et al. [41] also investigate the complexity of
729 reasoning with interpolation, and show that under both semantics the concept subsumption
730 problem becomes computationally more costly than in pure \mathcal{EL} : coNP -complete under the
731 feature semantics and PSPACE -hard under the geometric semantics.

732 One of the main drawbacks of the feature semantics is that it is too restrictive and cannot
733 support interpolation in an adequate way if the \perp construct is present. To address this
734 shortcoming, Schockaert et al. [62] recently introduced a new semantics based on an abstract
735 ternary betweenness relation bet over elements of the domain, such that that $\text{bet}(a, b, c)$ if
736 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
737 every element in $A^{\mathcal{I}}$ is between some individual from $B_1^{\mathcal{I}}$ and some element from $B_2^{\mathcal{I}}$. A
738 central result from [62] shows that the feature-enriched semantics from [41] can essentially
739 be seen as a special case, where the betweenness relation bet fulfills certain properties. The
740 results in [62] are preliminary, leaving open for example, the complexity of reasoning under
741 this new semantics.

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

747 following rules:

$$\begin{aligned}
 748 \quad & \text{burglary}(L, T) \leftarrow \text{burglary}(L, T - 2), \text{burglary}(L, T - 1) \\
 749 \quad & \text{burglary}(L, T) \leftarrow \text{burglary}(L, T - 1), \text{burglary}(L_1, T - 1), \text{burglary}(L_2, T - 1), n(L, L_1), \\
 750 \quad & \quad n(L, L_2), L_1 \neq L_2 \\
 751 \quad & \text{burglary}(L, T) \leftarrow \text{burglary}(L, T - 2), \text{burglary}(L_1, T - 1), \text{burglary}(L_2, T - 1), n(L, L_1), \\
 752 \quad & \quad n(L, L_2), L_1 \neq L_2 \\
 753
 \end{aligned}$$

754 Intuitively, these rules partially characterise the spatio-temporal diffusion pattern of crime
 755 hotspots. For instance, the first rule asserts that if there has been a burglary at time points
 756 $T - 1$ and $T - 2$ at a given location (e.g. during the two previous days), then it is likely that
 757 there will be a burglary at time point T in the same location. The other two rules include
 758 the predicate n to encode information about neighbouring locations. Given these rules, the
 759 following rule also seems plausible:

$$\begin{aligned}
 760 \quad & \text{burglary}(L, T) \leftarrow \text{burglary}(L_1, T - 2), \text{burglary}(L_2, T - 2), \text{burglary}(L, T - 1), n(L, L_1), \\
 761 \quad & \quad n(L, L_2), L_1 \neq L_2 \\
 762
 \end{aligned}$$

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

765 5.2 Analogical Reasoning

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

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

$$778 \quad \text{Young} \sqcap \text{Cat} \sqsubseteq \text{Cute} \tag{16}$$

$$779 \quad \text{Adult} \sqcap \text{WildCat} \sqsubseteq \text{Dangerous} \tag{17}$$

$$780 \quad \text{Young} \sqcap \text{Dog} \sqsubseteq \text{Cute} \tag{18}$$

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

$$785 \quad \text{Adult} \sqcap \text{Wolf} \sqsubseteq \text{Dangerous} \tag{19}$$

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

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

$$790 \quad \text{Program} \sqsubseteq \exists \text{specifies. Software} \quad (20)$$

792 and the fact that “Program is to Plan what Software is to Building”. Then we can plausibly
793 infer:

$$794 \quad \text{Plan} \sqsubseteq \exists \text{specifies. Building} \quad (21)$$

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

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

$$804 \quad C, D := \top \mid \perp \mid A \mid C \sqcap D \mid \exists r.C \mid N$$

$$805 \quad N, N' := A' \mid N \sqcap N' \mid N \bowtie N' \mid \exists r'.N$$

807 Note how $\mathcal{EL}_{\perp}^{\text{ana}}$ concepts extend $\mathcal{EL}_{\perp}^{\boxtimes}$ concepts by allowing existential restrictions over
808 so-called intra-domain roles, i.e. roles from the designated set $\mathbf{N}_{\mathbf{R}}^{\text{Int}}$, as natural concepts. An
809 $\mathcal{EL}_{\perp}^{\text{ana}}$ TBox is a finite set containing two types of assertions: (i) $\mathcal{EL}_{\perp}^{\text{ana}}$ concept inclusions,
810 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
811 $\mathcal{EL}_{\perp}^{\text{ana}}$ concepts.

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

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

825 **6 Conclusions**

826 Combining symbolic reasoning with sub-symbolic learning is an important and widely studied
827 challenge for AI research. To enable such a combination in a principled way, a key question
828 is how we can unify the two rather distinct types of representations that are involved, i.e.
829 symbols and vectors. In this paper, we discussed a number of strategies that are inspired
830 by the theory of conceptual spaces. First, we looked at the possibility of achieving a tight
831 integration between symbolic and vector representations based on the idea that concepts
832 can be viewed as regions in vector space embeddings. Moreover, we also explored the idea
833 that meaningful “quality dimensions” can be identified in learned embeddings, adding more

834 structure and a degree of interpretability to the vector representations themselves. However,
 835 we also argued that the use of region based representations has some inherent limitations
 836 when it comes to modelling relational knowledge. For this reason, we finally discussed a
 837 number of settings in which vectors and symbols are combined in a looser way. Essentially, the
 838 underlying idea is to exploit the similarity structure captured by the vector space to identify
 839 symbolic knowledge that plausibly, but not deductively, follows from a given knowledge base.

840 ——— References ———

- 841 1 Ralph Abboud, İsmail İlkan Ceylan, Thomas Lukasiewicz, and Tommaso Salvatori. BoxE: A
 842 box embedding model for knowledge base completion. In *NeurIPS*, 2020.
- 843 2 Thomas Ager. *Disentangling low-dimensional vector space representations of text documents*.
 844 PhD thesis, Cardiff University, 2021.
- 845 3 Thomas Ager, Ondrej Kuzelka, and Steven Schockaert. Modelling salient features as directions
 846 in fine-tuned semantic spaces. In *CoNLL*, pages 530–540, 2018.
- 847 4 Rana Alshaikh, Zied Bouraoui, Shelan S. Jeawak, and Steven Schockaert. A mixture-of-experts
 848 model for learning multi-facet entity embeddings. In *COLING*, pages 5124–5135, 2020.
- 849 5 Rana Alshaikh, Zied Bouraoui, and Steven Schockaert. Learning conceptual spaces with
 850 disentangled facets. In *CoNLL*, pages 131–139, 2019.
- 851 6 Rana Alshaikh, Zied Bouraoui, and Steven Schockaert. Hierarchical linear disentanglement of
 852 data-driven conceptual spaces. In *IJCAI*, pages 3573–3579, 2020.
- 853 7 Abhijeet Awasthi, Sabyasachi Ghosh, Rasna Goyal, and Sunita Sarawagi. Learning from rules
 854 generalizing labeled exemplars. In *ICLR*, 2020.
- 855 8 Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In *IJCAI*,
 856 pages 364–369, 2005.
- 857 9 Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description*
 858 *Logic*. Cambridge University Press, 2017.
- 859 10 Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. On rules with
 860 existential variables: Walking the decidability line. *Artif. Intell.*, 175(9-10):1620–1654, 2011.
- 861 11 Nelly Barbot, Laurent Miclet, and Henri Prade. Analogy between concepts. *Artificial*
 862 *Intelligence*, 275:487–539, 2019.
- 863 12 Sabri Bayouhd, Laurent Miclet, and Arnaud Delhay. Learning by analogy: A classification
 864 rule for binary and nominal data. In *IJCAI*, pages 678–683, 2007.
- 865 13 Islam Beltagy, Cuong Chau, Gemma Boleda, Dan Garrette, Katrin Erk, and Raymond J.
 866 Mooney. Montague meets Markov: Deep semantics with probabilistic logical form. In **SEM*,
 867 pages 11–21, 2013.
- 868 14 Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko.
 869 Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.
- 870 15 Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz. Recent advances in querying
 871 probabilistic knowledge bases. In *IJCAI*, pages 5420–5426, 2018.
- 872 16 Zied Bouraoui, José Camacho-Collados, Luis Espinosa Anke, and Steven Schockaert. Modelling
 873 semantic categories using conceptual neighborhood. In *AAAI*, pages 7448–7455. AAAI Press,
 874 2020.
- 875 17 Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. Inductive reasoning about ontologies
 876 using conceptual spaces. In *AAAI*, pages 4364–4370, 2017.
- 877 18 Zied Bouraoui and Steven Schockaert. Learning conceptual space representations of interrelated
 878 concepts. In Jérôme Lang, editor, *IJCAI*, pages 1760–1766, 2018.
- 879 19 Camille Bourgaux, Ana Ozaki, and Jeff Z. Pan. Geometric models for (temporally) attributed
 880 description logics. In Martin Homola, Vladislav Ryzhikov, and Renate A. Schmidt, editors,
 881 *DL*, 2021.

- 882 20 Katarina Britz, Thomas Meyer, and Ivan Varzinczak. Semantic foundation for preferential
883 description logics. In *Australasian Joint Conference on Artificial Intelligence*, pages 491–500.
884 Springer, 2011.
- 885 21 Andrea Cali, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering
886 under expressive relational constraints. *J. Artif. Intell. Res.*, 48:115–174, 2013.
- 887 22 José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: Integrating
888 explicit knowledge and corpus statistics for a multilingual representation of concepts and
889 entities. *Artificial Intelligence*, 240:36–64, 2016.
- 890 23 Claudia d’Amato, Nicola Fanizzi, Bettina Fazzinga, Georg Gottlob, and Thomas Lukasiewicz.
891 Ontology-based semantic search on the web and its combination with the power of inductive
892 reasoning. *Ann. Math. Artif. Intell.*, 65(2-3):83–121, 2012.
- 893 24 Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. Lifted rule injection for relation
894 embeddings. In *EMNLP*, pages 1389–1399, 2016.
- 895 25 Joaquín Derrac and Steven Schockaert. Inducing semantic relations from conceptual spaces:
896 A data-driven approach to plausible reasoning. *Artif. Intell.*, 228:66–94, 2015.
- 897 26 Didier Dubois, Henri Prade, Francesc Esteva, Pere Garcia, and Lluís Godó. A logical approach
898 to interpolation based on similarity relations. *International Journal of Approximate Reasoning*,
899 17(1):1–36, 1997.
- 900 27 Peter Gärdenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2000.
- 901 28 Peter Gärdenfors. How to make the semantic web more semantic. In A.C. Varzi and L. Vieu,
902 editors, *Formal Ontology in Information Systems*, pages 19–36. IOS Press, 2004.
- 903 29 Peter Gärdenfors. *The geometry of meaning: Semantics based on conceptual spaces*. MIT
904 press, 2014.
- 905 30 Peter Gärdenfors and Mary-Anne Williams. Reasoning about categories in conceptual spaces.
906 In *IJCAI*, pages 385–392, 2001.
- 907 31 Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*,
908 7(2):155–170, 1983.
- 909 32 Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. Semantic
910 characterization of rational closure: From propositional logic to description logics. *Artificial
911 Intelligence*, 226:1–33, 2015.
- 912 33 Robert L Goldstone. Isolated and interrelated concepts. *Memory & Cognition*, 24(5):608–628,
913 1996.
- 914 34 Nelson Goodman. *Fact, fiction, and forecast*. Harvard University Press, 1955.
- 915 35 Georg Gottlob, Michael Morak, and Andreas Pieris. Recent advances in datalog ^pm . In
916 *Reasoning Web*, volume 9203 of *Lecture Notes in Computer Science*, pages 193–217, 2015.
- 917 36 Víctor Gutiérrez-Basulto, Jean Christoph Jung, Carsten Lutz, and Lutz Schröder. Probabilistic
918 description logics for subjective uncertainty. *J. Artif. Intell. Res.*, 58:1–66, 2017.
- 919 37 Víctor Gutiérrez-Basulto and Steven Schockaert. From knowledge graph embedding to ontology
920 embedding? an analysis of the compatibility between vector space representations and rules.
921 In *KR*, pages 379–388, 2018.
- 922 38 Douglas R Hofstadter, Melanie Mitchell, et al. The copycat project: A model of mental fluidity
923 and analogy-making. *Advances in Connectionist and Neural Computation Theory*, 2:205–267,
924 1995.
- 925 39 Keith J Holyoak and Paul Thagard. The analogical mind. *American psychologist*, 52(1):35–44,
926 1997.
- 927 40 Nicolas Hug, Henri Prade, Gilles Richard, and Mathieu Serrurier. Analogical classifiers: A
928 theoretical perspective. In *ECAI*, pages 689–697, 2016.
- 929 41 Yazmín Ibáñez-García, Víctor Gutiérrez-Basulto, and Steven Schockaert. Plausible reasoning
930 about el-ontologies using concept interpolation. In *KR*, pages 506–516, 2020.
- 931 42 Sarit Kraus, Daniel Lehmann, and Menachem Magidor. Nonmonotonic reasoning, preferential
932 models and cumulative logics. *Artificial intelligence*, 44(1-2):167–207, 1990.

- 933 43 Tao Li and Vivek Srikumar. Augmenting neural networks with first-order logic. In *ACL*, pages
934 292–302, 2019.
- 935 44 Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De
936 Raedt. DeepProbLog: Neural probabilistic logic programming. In *NeurIPS*, pages 3753–3763,
937 2018.
- 938 45 Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space
939 word representations. In *NAACL-HLT*, pages 746–751, 2013.
- 940 46 Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation
941 and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*,
942 193:217–250, 2012.
- 943 47 Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective
944 learning on multi-relational data. In *ICML*, pages 809–816, 2011.
- 945 48 Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical represent-
946 ations. *NIPS*, 30:6338–6347, 2017.
- 947 49 Robert M Nosofsky. Choice, similarity, and the context theory of classification. *Journal of*
948 *Experimental Psychology: Learning, Memory, and Cognition*, 10(1):104–114, 1984.
- 949 50 Daniel N Osherson, Edward E Smith, Ormond Wilkie, Alejandro Lopez, and Eldar Shafir.
950 Category-based induction. *Psychological Review*, 97(2):185–200, 1990.
- 951 51 Matías Osta-Vélez and Peter Gärdenfors. Category-based induction in conceptual spaces.
952 *Journal of Mathematical Psychology*, 96, 2020.
- 953 52 Özgür Lütfü Özçep, Mena Leemhuis, and Diedrich Wolter. Cone semantics for logics with
954 negation. In *IJCAI*, pages 1820–1826, 2020.
- 955 53 Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for
956 word representation. In *EMNLP*, pages 1532–1543, 2014.
- 957 54 Henri Prade and Gilles Richard, editors. *Computational Approaches to Analogical Reasoning:*
958 *Current Trends*, volume 548 of *Studies in Computational Intelligence*. Springer, 2014.
- 959 55 Henri Prade and Gilles Richard. From analogical proportion to logical proportions: A survey.
960 In *Computational Approaches to Analogical Reasoning: Current Trends*, pages 217–244. 2014.
- 961 56 Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. Rnnlogic:
962 Learning logic rules for reasoning on knowledge graphs. In *ICLR*, 2020.
- 963 57 W.V. Quine. *From a Logical Point of View*. Harvard University Press, 1953.
- 964 58 Lance J Rips. Inductive judgments about natural categories. *Journal of Verbal Learning and*
965 *Verbal Behavior*, 14(6):665–681, 1975.
- 966 59 Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. In *NIPS*, pages
967 3788–3800, 2017.
- 968 60 Andrea Rossi, Denilson Barbosa, Donatella Firmani, Antonio Martinata, and Paolo Merialdo.
969 Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions*
970 *on Knowledge Discovery from Data*, 15(2):1–49, 2021.
- 971 61 Steven Schockaert, Yazmín Ibáñez-García, and Víctor Gutiérrez-Basulto. A description logic
972 for analogical reasoning. In *IJCAI*, pages 2040–2046. ijcai.org, 2021.
- 973 62 Steven Schockaert, Yazmín Angélica Ibáñez-García, and Víctor Gutiérrez-Basulto. Modelling
974 concept interpolation in description logics using abstract betweenness relations. In *DL*, 2021.
- 975 63 Steven Schockaert and Henri Prade. Solving conflicts in information merging by a flexible
976 interpretation of atomic propositions. *Artif. Intell.*, 175(11):1815–1855, 2011.
- 977 64 Steven Schockaert and Henri Prade. Interpolative and extrapolative reasoning in propositional
978 theories using qualitative knowledge about conceptual spaces. *Artif. Intell.*, 202:86–131, 2013.
- 979 65 Mikhail Sheremet, Dmitry Tishkovsky, Frank Wolter, and Michael Zakharyashev. A logic for
980 concepts and similarity. *Journal of Logic and Computation*, 17(3):415–452, 2007.
- 981 66 Steven A Sloman. Feature-based induction. *Cognitive Psychology*, 25(2):231–280, 1993.
- 982 67 Gustav Sourek, Vojtech Aschenbrenner, Filip Zelezný, Steven Schockaert, and Ondrej Kuzelka.
983 Lifted relational neural networks: Efficient learning of latent relational structures. *J. Artif.*
984 *Intell. Res.*, 62:69–100, 2018.

- 985 68 Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph
986 embedding by relational rotation in complex space. In *ICLR*, 2018.
- 987 69 Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge graph
988 embedding by relational rotation in complex space. In *ICLR*, 2019.
- 989 70 Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, and
990 Guillaume Bouchard. Knowledge graph completion via complex tensor factorization. *J. Mach.*
991 *Learn. Res.*, 18:130:1–130:38, 2017.
- 992 71 Asahi Ushio, José Camacho-Collados, and Steven Schockaert. Distilling relation embeddings
993 from pretrained language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia,
994 and Scott Wen-tau Yih, editors, *EMNLP*, pages 9044–9062, 2021.
- 995 72 Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey
996 of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*,
997 29(12):2724–2743, 2017.
- 998 73 Rudolf Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In
999 *Ordered Sets*, pages 445–470. 1982.
- 1000 74 Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss
1001 function for deep learning with symbolic knowledge. In *ICML*, pages 5498–5507, 2018.
- 1002 75 Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities
1003 and relations for learning and inference in knowledge bases. In *ICLR*, 2015.
- 1004 76 Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities
1005 and relations for learning and inference in knowledge bases. In *ICLR*, 2015.
- 1006 77 Zhun Yang, Adam Ishay, and Joohyung Lee. Neurasp: Embracing neural networks into answer
1007 set programming. In *IJCAI*, pages 1755–1762, 2020.
- 1008 78 LA Zadeh. Calculus of fuzzy restrictions. In *Fuzzy Sets and Their Applications to Cognitive*
1009 *and Decision Processes: Proceedings of the US–Japan Seminar on Fuzzy Sets and Their*
1010 *Applications, Held at the University of California, Berkeley, California, July 1-4, 1974*, pages
1011 1–39, 1975.
- 1012 79 Zhanqiu Zhang, Jie Wang, Jiajun Chen, Shuiwang Ji, and Feng Wu. ConE: Cone embeddings
1013 for multi-hop reasoning over knowledge graphs. *NeurIPS*, 2021.