

Conceptual Representations for Computational Concept Creation

PING XIAO, HANNU TOIVONEN, and OSKAR GROSS, University of Helsinki
 AMÍLCAR CARDOSO, JOÃO CORREIA, PENOUSAL MACHADO, PEDRO MARTINS,
 HUGO GONCALO OLIVEIRA, and RAHUL SHARMA, CISUC, University of Coimbra
 ALEXANDRE MIGUEL PINTO, LaSIGE, University of Lisbon
 ALBERTO DÍAZ, VIRGINIA FRANCISCO, PABLO GERVÁS, RAQUEL HERVÁS, and
 CARLOS LEÓN, Complutense University of Madrid
 JAMIE FORTH, Goldsmiths, University of London
 MATTHEW PURVER, Queen Mary University of London
 GERAINT A. WIGGINS, Vrije Universiteit Brussel, Belgium & Queen Mary University of London
 DRAGANA MILJKOVIĆ, VID PODPEČAN, SENJA POLLAK, JAN KRALJ, and
 MARTIN ŽNIDARŠIČ, Jožef Stefan Institute
 MARKO BOHANEC, NADA LAVRAČ, and TANJA URBANČIČ, Jožef Stefan Institute &
 University of Nova Gorica
 FRANK VAN DER VELDE, University of Twente
 STUART BATTERSBY, Chatterbox Labs Ltd.

Computational creativity seeks to understand computational mechanisms that can be characterized as creative. The creation of new concepts is a central challenge for any creative system. In this article, we outline different approaches to computational concept creation and then review conceptual representations relevant to concept creation, and therefore to computational creativity. The conceptual representations are organized in accordance with two important perspectives on the distinctions between them. One distinction is between

This work was supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733 (ConCreTe), and by the Academy of Finland under grants 276897 (CLiC) and 313973 (CACS).

Authors' addresses: P. Xiao, H. Toivonen, and O. Gross, Department of Computer Science, P.O. Box 68, FI-00014 University of Helsinki, Finland; emails: nurdito@gmail.com, hannu.toivonen@helsinki.fi, oskar.gross@gmail.com; A. Cardoso, J. Correia, P. Machado, P. Martins, H. Gonçalo Oliveira, and R. Sharma, CISUC, University of Coimbra, Polo 2, Pinhal de Marrocos, 3030-290 Coimbra, Portugal; emails: {amilcar, jncor, machado, pjmm, hroliv, rahul}@dei.uc.pt; A. M. Pinto, LaSIGE, University of Lisbon, Portugal; email: amdpinto@ciencias.ulisboa.pt; A. Díaz, V. Francisco, R. Hervás, P. Gervás, and C. León, Complutense University of Madrid, Spain; emails: {albertodiaz, virginia, raquelhb}@fdi.ucm.es, pgervas@sip.ucm.es, cleon@ucm.es; J. Forth, Goldsmiths, University of London, UK; email: j.forth@gold.ac.uk; M. Purver, Queen Mary University of London, UK; email: m.purver@qmul.ac.uk; G. A. Wiggins, Vrije Universiteit Brussel, Pleinlaan 9, B-1050 Brussel, Belgium; email: geraint@ai.vub.ac.be; M. Bohanec, N. Lavrač, D. Miljković, V. Podpečan, S. Pollak, J. Kralj, and M. Žnidaršič, Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia; emails: {marko.bohanec, nada.lavrac, dragana.miljkovic, vid.podpecan, senja.pollak, jan.kralj, martin.znidarsic}@ijs.si; T. Urbančič, University of Nova Gorica, Slovenia; email: tanja.urbancic@ung.si; F. V. D. Velde, Cognitive Psychology and Ergonomics, BMS, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands; email: f.vandervelde@utwente.nl; S. Battersby, Chatterbox Labs Ltd., UK; email: stuart@chatterbox.co.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0360-0300/2019/02-ART9 \$15.00

<https://doi.org/10.1145/3186729>

symbolic, spatial and connectionist representations. The other is between descriptive and procedural representations. Additionally, conceptual representations used in particular creative domains, such as language, music, image and emotion, are reviewed separately. For every representation reviewed, we cover the inference it affords, the computational means of building it, and its application in concept creation.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Knowledge representation and reasoning**; *Philosophical/theoretical foundations of artificial intelligence*; • **Applied computing** → *Arts and humanities*;

Additional Key Words and Phrases: Computational creativity, concept creation, concept, conceptual representation, procedural representation

ACM Reference format:

Ping Xiao, Hannu Toivonen, Oskar Gross, Amílcar Cardoso, João Correia, Penousal Machado, Pedro Martins, Hugo Goncalo Oliveira, Rahul Sharma, Alexandre Miguel Pinto, Alberto Díaz, Virginia Francisco, Pablo Gervás, Raquel Hervás, Carlos León, Jamie Forth, Matthew Purver, Geraint A. Wiggins, Dragana Miljković, Vid Podpečan, Senja Pollak, Jan Kralj, Martin Žnidaršič, Marko Bohanec, Nada Lavrač, Tanja Urbančič, Frank Van Der Velde, and Stuart Battersby. 2019. Conceptual Representations for Computational Concept Creation. *ACM Comput. Surv.* 52, 1, Article 9 (February 2019), 33 pages.

<https://doi.org/10.1145/3186729>

1 INTRODUCTION

Computational creativity (Wiggins 2006a; Colton and Wiggins 2012) is a field of research, seeking to understand how computational processes can lead to creative results. The field is gaining importance as the need for not only intelligent but also creative computational support increases in fields such as science, games, arts, music and literature. Although work on automated creativity has a history of at least 1,000 years (e.g., Guido d’Arezzo’s melody generation work around the year 1026), the field has only recently developed an academic identity and a research community of its own.¹

The computational generation of new concepts is a central challenge among the topics of computational creativity research. In this article, we briefly outline different approaches to concept creation and then review conceptual representations that have been used in various endeavors in this area. Our motivation stems from computational creativity, and our aim is to shed light on the emerging research area, as well as to help researchers in artificial intelligence more generally to gain an overview of conceptual representations, especially from the point of view of computational concept creation. For generic treatments of knowledge representation and reasoning, we refer the reader to other reviews (Brachman and Levesque 1985; Sowa 2000; Liao 2003; Van Harmelen et al. 2008; Jakus et al. 2013).

“Concept” has been used to refer to a bewildering range of things. In the pioneering study of creativity by Boden (1990), a concept is an *abstract idea* in arts, science, and everyday life. A key requirement of any computational model (creative or otherwise) that manipulates concepts, is for a *representation* of the concept being manipulated. Any representation describes some but not all aspects of an idea and supports a limited number of processing options (Davis et al. 1993), which are further constrained by the scientific or technical methods available at the time. A representation is created or chosen (over competing representations) based on the extent to which it facilitates a specific task. For an interdisciplinary audience, it is important to understand that the word *representation* used in different ways in different fields. In computer science, it means “a formalism for encoding data or meaning to be used within a computational system.” This is in contrast to the

¹<http://computationalcreativity.net/home/conferences/>.

use in psychology where a representation is a mental structure that corresponds with a specific thing or class of things; thus, for a psychologist, a concept and a representation are much the same thing. For us, however, one is a thing to be encoded using a representation.

This article reviews computational conceptual representations relevant to computational concept creation, which therefore may be useful in computational creativity. First, an overview of different approaches to concept creation is presented, providing a background to this review. The actual review that follows next is structured in terms of two major distinctions between different conceptual representations. One distinction is along the axis from connectionist to symbolic levels, with a spatial level between them. The other distinction, especially prominent in the computer science community, is between descriptive and procedural representations.

It is perhaps worth adding explicitly that the aim of this article is not to discuss the philosophy of cognitive concept creation (interesting though that be), but to survey relevant computational approaches in the field of computational creativity.

1.1 Approaches to Computational Concept Creation

In this section, we propose a typology of computational concept creation techniques as a background and motivation for the review of conceptual representations. Our aim here is to provide an overview of the multitude of approaches instead of giving a detailed review of concept creation techniques.

One of the best known categorizations of different types of creativity is by Boden (1990). Boden distinguishes combinatorial re-use of existing ideas, exploratory search for new ideas, and transformational creativity where the search space is also subject to change. Although Boden does not give a computational account of how concepts or ideas could be created, her taxonomy provides a starting point for characterizing various possible approaches to concept creation. We extend it with approaches based on extraction and induction. Our taxonomy of different types of concept creation approaches is reflected in the different kinds of input they take: concept extraction transforms an existing representation of a concept to a different representation, concept induction generalizes from a set of instances of concepts, concept recycling modifies existing concepts to new ones, and concept space exploration takes a search space of concepts as input:

- *Concept extraction* is the task of extracting and transforming a conceptual representation from an existing but different representation of the same idea. Typically, the extracted representation is more concise, explicit, and centralized. Concept extraction is frequently applied to textual corpora, such as extracting semantic relations between concepts. For instance, the conceptual information that wine is a drink could be extracted from the natural language expression “wine and other drinks.” For an overview of information extraction methods, see, for example, the work of Sarawagi (2008).
- *Concept induction* is based on instances of concepts from which a concept or concepts are learned. Drawing from the field of machine learning and data mining, two major types of inductive concept creation can be identified:
 - *Concept learning* is a supervised activity where a description of a concept is formed inductively from (positive and negative) examples of the concept (e.g., Kotsiantis (2007)). In logical terms, (a sample from) the extension of a concept is given and machine learning is then used to construct its intension. An example from the field of music is a system that is given songs by the Beatles and other bands, and the system then learns the typical chord progression patterns, as concepts, used by the Beatles.
 - *Concept discovery* is an unsupervised inductive activity where the memberships of instances in concepts are not known in advance, but methods such as clustering are used

to generalize natural concepts from given examples (e.g., Jain et al. (1999)). Continuing the preceding example, given songs by a number of bands, the system can discover different genres of music by clustering the songs.

Although inductive methods can be used to uncover existing concepts, more interesting applications arise when these methods are used to discover and formulate new possible concepts based on data.

- *Concept recycling* is the creative re-use of existing concepts (e.g., refer to Aamodt and Plaza (1994)). (The original concepts usually stay in existence as well; recycling does not imply that they have to be removed.) We mention two typical methods:
 - *Concept mutation*, in which some given concept is modified by adding, removing, or changing something in it. For instance, the concept of “mobile phone” has been modified to “smart phone” by allowing new applications to be used on the phones. Changing existing concepts is a well-known form of concept creation also with humans; Osborn (1953) lists a variety of techniques for such purpose. Two common mutational operations are *generalization* and *specialization* of existing concepts.
 - *Concept combination*, when two or more existing concepts are combined to form a new one, such as in *conceptual blending*, where “computer program” and “virus” can form a conceptual blend of “computer virus.”

Both mutation and (re-)combination of concepts are utilized heavily in evolutionary methods for concept creation. Although recycling may sound like an easy way to create new concepts, a key problem is to measure how meaningful and interesting the generated concepts are.

- *Concept space exploration* takes as input a search space of possible new concepts and locates interesting concepts in it. The space can be specified either declaratively or procedurally. A poetry writing system, for instance, can take as input a template with blanks to fill with words. This specifies a search space of possible poems, or concepts. Again, a crucial and non-trivial issue is how the quality of new concepts is measured.

The preceding accounts of concept creation differ in terms of their input and thus help to characterize different settings where concept creation is used. The techniques involved may, however, share methodological similarities; refer to the references given previously for overviews. In particular, many of the preceding concept creation techniques can actually be described as search (Wiggins 2006a, 2006b), where the search space and the ways of traversing it depend on the specific technique. This makes it sometimes difficult to tell if a system is exploratory in nature or rather more of one of the preceding types of concept creation. For instance, different operations used in concept mutation can be seen as traversing a space of concepts; the space reachable by the system is defined by the initial concept to be mutated and all possible combinations of mutation operations.

Additionally, there is *transformational* creativity where the system also adjusts its own operation (Boden 1990). Transformational or meta-creativity takes any of the preceding types of concept creation onto a more abstract level where additionally the data, assumptions, search methods, evaluation metrics, or goals are also modified. This potentially takes a system toward higher creative autonomy (Jennings 2010).

The formalization by Wiggins (2006a) of creativity as search gives a unifying formalization over Boden’s categorization (and of ours). He shows not only how both combinatorial and exploratory creativity can be seen as search at the concept level, as already mentioned, but also how transformational creativity can be seen as search at the meta-level (i.e., on a level including also meta-information about the concept level search). This provides a powerful way of describing and comparing a wide range of creative systems.

As can be seen from the preceding categorization, techniques from machine learning find applications in several concept creation tasks. For a recent review on the use of machine learning in computational creativity, including the transformational case, refer to Toivonen and Gross (2015).

The space does not allow proper review of approaches and techniques suitable for concept creation, so that is left outside the scope of this article. In the review that follows, we complement the more abstract works of Boden (1990) and Wiggins (2006a) by presenting concrete conceptual representations that have been used in computational concept creation.

1.2 Organizing Conceptual Representations

Conceptual representations can be distinguished, at least, under two perspectives. One is the *level* of representation (symbolic, spatial or connectionist representations), and the other is descriptive or procedural representations. We give a brief introduction to these distinctions next.

Levels of conceptual representations. Gärdenfors (2000) proposes three levels of cognitive representations: a symbolic level; a conceptual level modeled in terms of *conceptual spaces*, which we term the *spatial level* here; and a sub-conceptual connectionist level. His theory is aimed at studying cognitive functions of humans (and other animals), as well as artificial systems capable of human cognitive functions. The idea is that all of these three levels are connected, sensory input to the connectionist level feeding spatial representations of concepts, which then become symbolic at the level of language:

- At the *symbolic level*, information is represented by *symbols*. *Rules* are defined to manipulate symbols. Symbolic representations are often associated with Good Old Fashioned AI² (GOFAI) (Haugeland 1985). An underlying assumption of GOFAI research is that human thinking can be understood in terms of symbolic computation, particularly computation based on formal principles of logic. However, symbolic systems have proved less successful in modeling aspects of human cognition beyond those closely related to logical thinking, such as perception. Furthermore, within a symbolic representation, meaning is internal to the representation itself; symbols have meaning only in terms of other symbols and not directly in terms of any real-world objects or phenomena they may represent. Gärdenfors proposes addressing this *symbol grounding problem* by linking symbols at this level to conceptual structures at the spatial conceptual level below. In linguistic terms, words (or expressions in a language of thought (Fodor 1975)) exist at this symbolic level but ground their meaning (semantics) in the spatial level.
- At the *spatial level*, information is represented by points or regions in a conceptual space that is built on *quality dimensions* with defined geometrical, topological, or ordinal properties. Similarity between concepts is represented in terms of the distance between points or regions in a multidimensional space. This formalism offers a parsimonious account of concept combination and acquisition, both of which are closely related to conceptual similarity. Moreover, by defining dimensions with respect to perceptual qualities, spatial representations are grounded in our experience of the physical world, which provides a semantics closely aligned with a human sense of meaning.
- At the *connectionist level*, information is represented by activation patterns in densely connected networks of primitive units. A particular strength of connectionist networks is their ability to learn concepts from observed data by progressively changing connection weights. Nevertheless, the weights between units in the network offer limited explanatory insights into the process being modeled, which requires an understanding of the computation of each unit.

²https://en.wikipedia.org/wiki/Symbolic_artificial_intelligence.

The three levels of representation outlined previously differ in representational granularity, and each level has its own strengths and weaknesses in modeling cognitive and creative functions. Although Gärdenfors (2000) takes the spatial level (which he refers to as the conceptual level) as the most appropriate for modeling concepts, he stresses the links between the levels and that different representational formalisms should be seen as complementary rather than competing. As such, choices of representation should be made in accordance with scientific aims and in response to the challenges of the particular problem at hand.

It is important to distinguish between the axis of these different levels of computational representation and the quite orthogonal axis supplied by the nature of human conceptualization, most particularly shown in categorical perception. Here, the hierarchical form is of the thing being modeled and not of the modeling formalism.

For example, a specific phenomenon that can occur in human conceptualization is found in categorical perception, in which the process of categorization has an amplifying effect on the underlying perception, by strengthening the perceptual differences across category boundaries and weakening them within boundaries. The classical example is speech perception, in which a varying speech sound is identified in terms of either one or another distinctive phoneme, even though a sharp distinction is objectively not present in the underlying sound pattern. Other examples of categorical perception are found in music (e.g., musical note identification) and vision (color categorization)—refer to Goldstone and Hendrickson (2010) for a recent overview. A further complication, as for example in color categorization, is that the categories can be hierarchical, as “red” is a super-category of “scarlet” and “vermilion.” The representations surveyed here capture these points in various ways.

In addition, in the case of human conceptualization, the question arises as to why (e.g., from an evolutionary or developmental perspective) concept formation based on perceptual categorizations emerged—in other words, why humans would tend to distinguish different categories even when the underlying perceptual domain is continuous. Perhaps this results from the close (evolutionary) relation between perception and action (van der Velde 2015). In performing an action, we (often) need to choose just one option out of many, given the physical boundary conditions related to the action. Simply stated, we can run in only one direction at a time, which forces a choice between the many options that could be available. The process of making such a choice could also affect the process of classifying the underlying perceptual domain.

Descriptive or procedural representations. The other perspective to conceptual representations is the distinction between descriptive and procedural representations. As the name indicates, a *descriptive representation* describes the artifact being represented. The description may be low or high level, complete, or partial. A *procedural representation*, however, specifies a procedure (e.g., a program) that once executed produces the artifact being represented. Similar to descriptive representations, the procedure may be low or high level, complete, or partial. A procedural representation is sometimes more succinct than a descriptive representation of the same idea. An example is the Fibonacci numbers—comparing the definition based on recurrence with an infinite sequence of integers. Conversely, for everything we know how to produce, there exists at least one procedural representation, although in many cases descriptive representations are employed. Like the three levels of representations introduced earlier, descriptive and procedural representations each have particular advantages and deficiencies. At each of the three levels, both descriptive and procedural representations exist or can be constructed.

A noteworthy point is that the conceptual representations discussed in this review are for the purpose of eliciting certain information or knowledge that affords certain inferences. Any conceptual representation can be “represented” in other forms for purposes other than the one discussed here (e.g., any procedural representation has to be implemented in program code to be executed).

Outline and structure of the article. In this review, we bring together conceptual representations relevant to concept creation, which are either the targets of concept creation or part of creating other concepts. Some representations have general presence in computer science, whereas others were especially created for concept creation, such as bisociation (Section 2.1), procedural representations of music and image (Section 5), and plan operator for story generation (Online Supplement). A part of the conceptual representations reviewed is relevant to a broad range of creative tasks (Sections 2, 3, 4, and 5), whereas a part of them is unique for certain creative domains (Online Supplement). For every representation reviewed, we cover the inference it supports, the computational means of building it, and its application in concept creation.

The conceptual representations included in this review are primarily organized according to the distinction between symbolic, spatial and connectionist representations, in Sections 2, 3, and 4 respectively. Symbolic and spatial representations are predominately descriptive, and connectionist representations can largely be considered procedural. Procedural representations, across the three levels, are reviewed in Section 5. The Online Supplement introduces conceptual representations used in four popular research domains of the computational creativity community: language, music, image, and emotion. The information in the four domains may have representations at all three levels and both descriptive and procedural representations. Discussions on the results of this review, conclusions, and future work are presented in Section 6.

2 SYMBOLIC REPRESENTATIONS

Symbols are used to represent objects, properties of objects, relationships, ideas, and so forth. Certain symbols might be better discussed within domains. For instance, in the Online Supplement, we introduce symbols used in the domains of language, music, image, and emotion, such as word, music note, and pixel matrix. These are atomic representations. Examples of more complex symbolic representations include plan operator, Scalable Vector Graphics (SVG) file, and emotional categories. In this section, we present symbolic representations that are applicable to many domains, including association, semantic relation, semantic network, ontology, information network, and logic.

2.1 Association

Association means “something linked in memory or imagination with a thing or person; the process of forming mental connections or bonds between sensations, ideas, or memories.”³ Association assumes a connection between concepts or symbols $C1$ and $C2$ but does not assume any specific conditions on $C1$ and $C2$. In addition, the nature of the connection is not of primary focus, contrasting it with semantic relation (Section 2.2), semantic network (Section 2.3), and ontology (Section 2.4).

In the field of computational creativity, the associations spanning over two different contexts (domains/categories/classes) are of special interest, in line with Mednick’s definition of creative thinking as the ability of generating new combinations of *distant associative elements* (Mednick 1962). Koestler (1964) also identified such cross-domain associations as an important element of creativity and calls them *bisociations*. Figure 1 shows a schematic representation of bisociation, where concepts $C1$ and $C2$, from two different contexts, $D1$ and $D2$, respectively, are bisociated. Bisociations, in various contexts, may be especially useful in discovering or creating analogies and metaphors.

According to Berthold (2012), bisociation can be informally defined as “(sets of) concepts that bridge two otherwise not—or only very sparsely—connected domains whereas an association bridges concepts within a given domain.” We argue that two concepts are bisociated if there is no

³<http://www.merriam-webster.com/dictionary/association>.

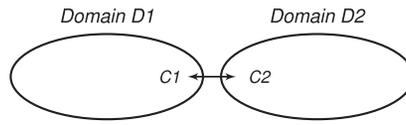


Fig. 1. Bisociation between concept $C1$ in domain $D1$ and concept $C2$ in domain $D2$.

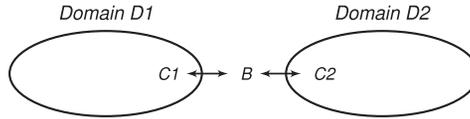


Fig. 2. Bisociation between concepts $C1$ and $C2$ using a bridging concept B .

direct, obvious evidence linking them, one has to cross contexts to find the link, and this new link provides novel insights into the domains. The bisociative connection between $C1$ and $C2$ may be represented together with a *bridging* concept B , which has links to both $C1$ and $C2$ (Figure 2). An example is the bisociation between evolution in nature and evolutionary computing bridged with the concept “optimization.” In addition to bridging concepts, Berthold (2012) introduces other types of bisociations: *bridging graphs* and *bridging by structural similarity*. The author points out that bridging concepts and bridging graphs require that the two domains have a certain type of neighborhood relation, whereas bridging by structural similarity allows matching on a more abstract level.

A number of bisociation discovery methods are based on graph representations of domains and finding cross-domain connections that are potentially new discoveries (Dubitzky et al. 2012). It has also been shown by Swanson (1990) that bisociation discovery can be tackled using literature mining methods. Swanson proposed a method for finding hypotheses spanning over previously disjoint sets of literature. To find out whether phenomenon a is associated with phenomenon c although there is no direct evidence for this in the literature, he searches for intermediate concepts b connected with a in some articles, and with c in some others. Putting these connections together and looking at their meaning may provide new insights about a and c .

Let us illustrate this with an example that has become well known in literature mining. In one of his studies, Swanson [1990] investigated if magnesium deficiency could cause migraine headaches. He found more than 60 pairs of articles—consisting of one article from the literature about migraine (c) and one article from the literature about magnesium (a)—connecting a with c via various third terms b . For example, in the literature about magnesium, there is a statement that magnesium is a natural calcium channel blocker, whereas in the literature about migraine, we read that calcium channel blockers can prevent migraine attacks. In this case, calcium channel blockers are a bridge between the domains of magnesium and migraine. Closer inspection showed that 11 identified pairs of documents were, when put together, suggestive and supportive for a hypothesis that magnesium deficiency may cause migraine headaches (Swanson 1990).

Many researchers have followed and further developed Swanson’s idea of searching for linking terms between two domains in the framework of literature mining. An overview of the literature-based discovery approaches and challenges is provided by Bruza and Weeber (2008). Furthermore, some other data mining approaches, such as co-clustering and multi-mode clustering (Govaert and Nadif 2014), may be suitable for identifying certain types of bisociations.

2.2 Semantic Relation

Broadly speaking, semantic relations are labeled relations between meanings, as well as between meanings and representations. In contrast to associations, semantic relations have meanings as

Table 1. Selected Research on Extracting Semantic Relations

Hyponymy-hypernymy	Hearst 1992; Caraballo 1999; Kozareva et al. 2008; Pantel and Ravichandran 2004; Pantel and Pennacchiotti 2006; Snow et al. 2005; Navigli and Velardi 2010; Bordea et al. 2015
Meronymy-holonymy	Berland and Charniak 1999; Girju et al. 2006; Ittoo et al. 2010; Pantel and Pennacchiotti 2006
Causal relations	Khoo et al. 2000; Girju and Moldovan 2002; Blanco et al. 2008; Ittoo and Bouma 2011

indicated by their labels. The number of semantic relations is virtually unlimited. Some important semantic relations are *synonymy*, *homonymy*, *antonymy*, *hyponymy-hypernymy*, *meronymy-holonymy*, *instance_of relation*, *causal relation*, *locative relation*, and *temporal relation*. In addition to the general relation types, there are domain-specific relations, such as the *ingredient_of* relation in the food domain or the *activates* relation in the biomedicine domain.

Here we briefly review how semantic relations have been extracted from various sources, mostly text (Table 1 provides a summary of selected research). We distinguish between approaches based on lexico-syntactic patterns and machine learning. The pioneering work of Hearst (1992) opened the era of discovering semantic relations using lexico-syntactic patterns. An example of lexico-syntactic patterns used in the automatic detection of hyponyms is “NP₁ such as NP₂,” where NP₂, a noun phrase, is potentially a hyponym of NP₁, another noun phrase. With a set of seed instances of certain relation (e.g., hyponymy), this method identifies sequences of text that occur systematically between the concepts of the instances. The patterns discovered are used in the automatic extraction of new instances. In this line of work, the relations that form the backbone of ontologies were first considered, such as hyponymy (Hearst 1992) and meronymy (Berland and Charniak 1999), followed by other relations, such as book author (Brin 1999), organization location (Agichtein and Gravano 2000), and inventor (Ravichandran and Hovy 2002).

As an alternative approach, machine learning techniques have been applied to the extraction of semantic relations from text. An overview of methods with different degrees of supervision is given by Nastase et al. (2013). Unsupervised methods, such as those based on clustering or co-occurrence, are mostly used to discover hypernymy and synonymy relations, and often in combination with pattern-based methods (Caraballo 1999; Pantel and Ravichandran 2004). In supervised machine learning, models are trained by generalizing from labeled examples of expressed relations. Labeled data was provided as part of some shared tasks, such as semantic evaluation workshops (SemEval). SemEval-2007 Task 4 (Girju et al. 2007) provides a dataset of meronymy, causality, origin, and so forth, followed by a dataset of some other relations in SemEval-2010 Task 8 (Hendrickx et al. 2010). In addition, SpaceEval 2015⁴ focuses on various spatial relations, such as path, topology, and orientation. Navigli and Velardi (2010) used an annotated dataset of definitions and hypernyms for learning word class lattices. Instead of manually annotated datasets, WordNet (Fellbaum 1998; Miller et al. 1990), Wikipedia,⁵ and other resources can be used for distant supervision (as large seeds for bootstrapping) (Snow et al. 2005; Mintz et al. 2009).

In contrast to extracting predefined semantic relations, the Open Information Extraction (OIE) paradigm does not depend on predefined patterns but considers relations as expressed by parts of speech (Fader et al. 2011), paths in a syntactic parse tree (Ciaramita et al. 2005), or sequences

⁴<http://alt.qcri.org/semEval2015/task8/>.

⁵<http://www.wikipedia.org>.

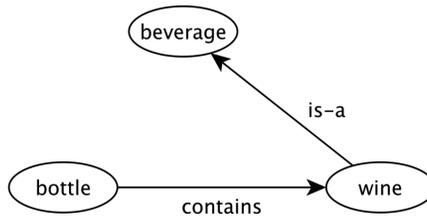


Fig. 3. Example of a semantic network.

of high-frequency words (Davidov and Rappoport 2006). OIE methods are used in ReVerb (Etzioni et al. 2011), Ollie (Mausam et al. 2012), and ClausIE (Del Corro and Gemulla 2013).

In addition to general semantic relations, automatically extracted semantic relations have been part of knowledge discovery in many specific domains, such as biology (Miljkovic et al. 2012), and have potential for concept creation as well. Semantic relations are also important components of semantic networks (Section 2.3) and ontologies (Section 2.4).

As well as the specific relations between symbols discussed here, other approaches take a more generally relational view of semantics, seeing text or word meanings in terms of statistical relations to other words (e.g., in the form of topic models, latent vectors, or distributional semantics); we introduce such approaches in the discussion about spatial representations (Section 3).

2.3 Semantic Network

Semantic networks (Sowa 1992) are a category of symbolic representations that represent collections of semantic relations between concepts. Figure 3 shows a small semantic network, which represents the sentences “The bottle contains wine. Wine is a beverage.” The concepts (i.e., “bottle,” “wine,” and “beverage”) are denoted by nodes, and the relations between them (i.e., *contains/contain* and *is-a*) are represented by directed edges. The meaning of a concept is defined in terms of its connections with other nodes (concepts). The closely related semantic link networks (Zhuge 2012) are self-organized semantic models similar in many ways to semantic networks but emphasizing larger semantic richness and automatic link discovery.

An example of existent large semantic networks is ConceptNet (Liu and Singh 2004), a semantic network of commonsense knowledge. In ConceptNet, nodes are semi-structured English fragments, including nouns, verbs, adjectives, and prepositional phrases. Nodes are interrelated by 1 of the 24 types of semantic relations, such as *IsA*, *PartOf*, *UsedFor*, *MadeOf*, *Causes*, *HasProperty*, *DefinedAs*, and *ConceptuallyRelatedTo*, represented by directed edges. The early versions of ConceptNet were built on the data of the Open Mind Common Sense Project,⁶ which collects commonsense knowledge from volunteers on the Web by asking them to fill in the blanks in sentences (Singh et al. 2002). ConceptNet 5,⁷ the current version, extends the previous versions with information automatically extracted from Wikipedia, Wiktionary,⁸ and WordNet.

Semantic networks have wide applications, such as database management (Roussopoulos and Mylopoulos 1975), cybersecurity (AlEroud and Karabatis 2013), and software engineering (Karabatis et al. 2009). Semantic networks are also popular knowledge sources in the computational creativity community. ConceptNet alone has been used in generating cross-domain analogies (Baydin et al. 2012), metaphor ideas for pictorial advertisements (Xiao and Blat 2013), Bengali poetry (Das and Gambäck 2014), and fictional ideas (Llano et al. 2016), as well as testing the

⁶<http://openmind.media.mit.edu>.

⁷<http://conceptnet5.media.mit.edu>.

⁸http://en.wiktionary.org/wiki/Wiktionary:Main_Page.

novelty of visual blends (Martins et al. 2015). Baydin et al. (2012) actually go beyond just using semantic networks—they also generate novel analogous semantic networks using an evolutionary algorithm.

Formally, semantic networks can in some cases be viewed as alternatives to spatial representations, the most obvious case being where nodes correspond with points in the space, and relations attached to arcs correspond with distances between them. This, however, is not a very conventional view. More common is the usage where nodes represent objects and values, and arcs represent relations between things represented by nodes, which is a much less subtle and more logic-like approach. The geometry of spatial representations affords many implicit concepts that must be made explicit in the network formalism; this may be positive or negative depending on the application. For example, in color space, red is a super-concept of scarlet and vermilion merely by virtue of geometry, and, given that the space is a good perceptual model, distances are implicit and do not need to be recorded; in a network representation, the super/sub-concept relation would need to be recorded explicitly. But many concepts do not conform to the regularity of Euclidean geometry, and in these cases, a network representation may be more appropriate.

As a symbolic representation, the meanings of concepts (nodes) in a semantic network are specified purely in terms of relations to other symbolic concepts—there is no grounding (Harnad 1990). However, mappings between symbolic networks and spatial representations, such as the Gärdenfors (2000) theory of conceptual space (cf. Section 3.1), could be developed to leverage the strengths of each form of representation. For example, the “wine” and “beverage” concepts from the semantic network in Figure 3 could correspond to regions in a conceptual space, whereby “wine” would typically be a sub-region of “beverage” within some set of contextually appropriate dimensions. This geometrical relationship implicitly represents that “wine” is a kind of “beverage,” as opposed to the explicit “is-a” type relation used in the semantic network.

2.4 Ontology

A widely accepted definition of ontology in computer science is by Gruber (2009):

“In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members).”

In computer science (and computational creativity), we can thus understand ontologies as resources of formalized knowledge, but the degree of formalization can vary.⁹

The relationship between ontologies and semantic networks is that ontologies provide representational primitives that can be used in semantic networks. The structural backbone of an ontology is a *taxonomy*, which defines a hierarchical classification of concepts, whereas an ontology represents a structured knowledge model with various kinds of relations between concepts and, possibly, rules and axioms (Navigli 2016). For instance, the expression “the bottle contains wine” in a semantic network (Figure 3) obtains a much richer meaning when combined with ontologies that include “bottle,” “contains,” and “wine,” as well as their properties and relations allowing one to reason with them.

We can distinguish between upper, middle, and domain ontologies (Navigli 2016). *Upper ontologies* encode high-level concepts (e.g., concepts like “entity,” “object,” and “situation”) and are usually constructed manually. Their main function is to support interoperability between domains.

⁹Different communities use the word *ontology* for different meanings. In philosophy, ontology refers to the philosophical discipline dealing with the nature and structure of “reality” and is opposed to epistemology, which concerns the understanding of reality and the nature of knowledge (Guarino and Giarretta 1995).

They are often linked to several lower-level ontologies. Examples of upper-level ontologies are SUMO (Pease and Niles 2002), DOLCE (Gangemi et al. 2002), and Cyc¹⁰ (Lenat 1995). *Middle ontologies* are general-purpose ontologies and provide the semantics needed for attaching to domain-specific concepts. For instance, WordNet (Fellbaum 1998; Miller et al. 1990) is a widely used middle ontology. *Domain ontologies* model the concepts, individuals, and relations of the domain of interest (e.g., the Gene Ontology¹¹). Existent ontologies often have more than one level. For example, SUMO and WordNet contain the upper-level concepts, middle ontologies, and some domain ontologies.

From the perspective of the amount of conceptualization, Sowa (2010) and Biemann (2005) distinguish between formal (also called *axiomatized*), terminological (also called *lexical*), and prototype-based ontologies (Figure 4):

- *Formal ontologies* (e.g., SUMO) are represented in logic, using axioms and definitions. Their advantage is the inference mechanism, enabling the properties of entities to be derived (in Figure 4, it is shown how one can derive that “chili con carne” is “non-vegetarian” food). Nevertheless, a high encoding effort is needed, and there is a danger of running into inconsistencies.
- *Terminological ontologies* (e.g., WordNet) have concept labels (terms used to express them in natural language). The lexical relations between terms, such as synonymy, antonymy, hypernymy, and meronymy, determine the relative positions of concepts but do not completely define them. The difference between a terminological ontology and a formal ontology concerns its degree (Sowa 2010). If represented as a graph, a terminological ontology is a special type of semantic network (see Section 2.3).
- *Prototype-based ontologies* represent categories (concepts) with typical instances (rather than concept labels or axioms and definitions in logic). New instances are classified based on a selected measure of similarity. Typically, prototype-based ontologies are taxonomies, because they are limited to hierarchical (unspecified) relations and are constructed by clustering techniques. Formal and prototype-based ontologies are often combined into mixed ontologies, where some sub-types are distinguished by axioms and definitions, but other sub-types are distinguished by prototypes. The theory of Gärdenfors (2000), which affords a kind of geometrical reasoning comparable with the logical reasoning afforded by ontologies, also affords reasoning about prototypes.

Taxonomy and ontology learning can be built on the methods of automatically extracting semantic relations (see Section 2.2). For instance, Kozareva and Hovy (2010) and Velardi et al. (2013) use a combination of pattern- and graph-based techniques. In addition, clustering techniques have been used in taxonomy learning (Cimiano et al. 2005; Fortuna et al. 2006).

Ontologies have been used in many computational creativity tasks. The combination of thematically different ontologies is applied in modeling analogies (relating different symbols based on their similar axiomatisation), metaphors (blending symbols of a source domain into a target domain, based on an analogy, and imposing the axiomatization of the former on the latter), pataphors (extending a metaphor by blending additional symbols and axioms from the source domain into the target, thus resulting in a new domain where the metaphor becomes reality), and conceptual blending (blending and combining two domains for the creation of new domains) (Kutz et al. 2012).

¹⁰OpenCyc is a public version of Cyc (which is proprietary).

¹¹<http://geneontology.org>.

a) **Formal ontology**

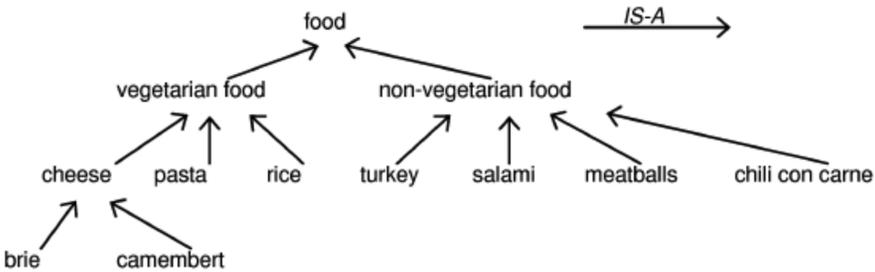
Axioms:

food(brie), food(camembert), food(turkey), food(meatballs), food(chili con carne), meat(turkey), meat(minced meat), part_of(minced meat, chili con carne), part_of(minced meat, meatballs)

veg_food(x) = { x | food(x) ∧ (¬part_of(y,x) ∧ meat(y)) ∧ ¬meat(x) }
 non_veg_food(x) = { x | food(x) ∧ ((part_of(y,x) ∧ meat(y)) ∨ meat(x)) }

Possible to derive: "turkey" and "chili con carne" are non-vegetarian foods

b) **Terminological ontology**



c) **Prototype-based ontology**

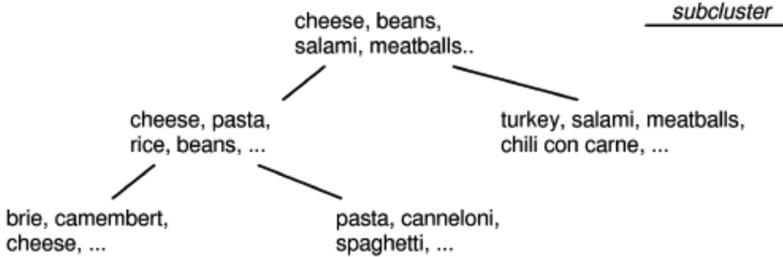


Fig. 4. Examples of formal ontology (a), terminological ontology (b), and prototype-based ontology (c) (Biemann 2005). Note that (b) and (c) illustrate only the structural backbones of the two ontologies. © Chris Biemann. Reproduced by permission.

2.5 **Information Network**

Information networks refer to any structure with connected entities, such as social networks. Mathematically, an information network can be represented as a graph, where graph vertices represent the entities, and edges represent the connections between them. Semantic networks are a special case of information networks where vertices and edges carry semantic meaning (i.e., labeled by semantic concepts). Information networks represent a broader category of knowledge representation than semantic networks. Study of information networks is less focused on the meaning encoded in the connections (which is the main focus of studying semantic networks) but more on the structure of networks

Studies of information networks include the work of Sun and Han (2013), where an information network is defined simply as a directed graph where both the nodes and edges have types, and the edge type uniquely characterizes the types of its adjacent nodes. When there is more than one

type of node or edge in an information network, the network is called a *heterogeneous information network*; if it has only one type of node and only one type of edge, it is a *homogeneous information network*.

There are plenty examples of information networks. Bibliographic information networks (Juršič et al. 2012; Sun and Han 2012) are networks connecting the authors of scientific papers with their papers. Specifically, they are heterogeneous networks with two types of nodes (authors and papers), and two types of edges (citations and authorships). Online social networks represent the communication in online social platforms. Biological networks contain biological concepts and the relations between them.

The methods of discovering new knowledge in homogeneous information networks can be split into several categories: node/edge label propagation (Zhou et al. 2003), link prediction (Barabási et al. 2002; Adamic and Adar 2003), community detection (Yang et al. 2010; Plantié and Crampes 2013), and node/edge ranking (Jeh and Widom 2002; Kondor and Lafferty 2002). A popular set of methods is based on eigenvalues and eigenvectors (commonly referred to as spectral methods). For example, in detecting communities, the community structure is extracted from either the eigenvectors of the Laplacian matrix (Donetti and Munoz 2004) or the stochastic matrix (Capocci et al. 2005) of the network.

The methods developed for homogeneous information networks, as introduced earlier, can be applied to heterogeneous information networks by simply ignoring the heterogeneous information altogether. This does, however, decrease the amount of information used and can therefore decrease the performance of the algorithms (Davis et al. 2011). Approaches that take into account the heterogeneous information are therefore preferable, such as network propositionalization (Grčar et al. 2013), authority ranking (Sun et al. 2009; Sun and Han 2012), ranking-based clustering (Sun et al. 2009; Sun and Han 2012), classification through label propagation (Hwang and Kuang 2010; Ji et al. 2010), ranking-based classification (Sun and Han 2012), and multi-relational link prediction (Davis et al. 2011).

2.6 Logic

Many kinds of logics have been used to represent concepts and complex knowledge, such as classical first-order logic, modal logics—including linear temporal logic and deontic logic—and other non-classical logics (e.g., default and non-monotonic logics).

A declarative representation of a concept can be a single symbol. More complex concepts can be represented by the composition of simpler formulas (corresponding to simpler concepts). These compositions are built by establishing some relations (e.g., conjunction, disjunction, negation, implication) between concepts. Ontologies (see Section 2.4) are built with a specific sub-family of logic languages, such as the description logics—the *veg_food(x)* concept in Figure 4 is one such example resorting to first-order logic.

Logic-based symbolic approaches can be used to represent and reason with both time-independent and temporal concepts (Bouzid et al. 2006). In addition to descriptive representations of concepts, logic-based approaches, particularly logic programs, can also be used for symbolic procedural representations via inductive definitions (Hou et al. 2010). For example, the following logic program (written in Prolog) defines the concepts of even and odd natural numbers, assuming that *suc(X)* stands for the *successor* of the natural number *X*:

$$\begin{aligned} & \text{even}(0). \\ \text{even}(\text{suc}(X)) & :- \text{odd}(X). \\ \text{odd}(\text{suc}(X)) & :- \text{even}(X). \end{aligned}$$

The use of logical representations and tools in computational creativity tasks has just started. Two of such works concern the computational modeling of conceptual blending, a cognitive

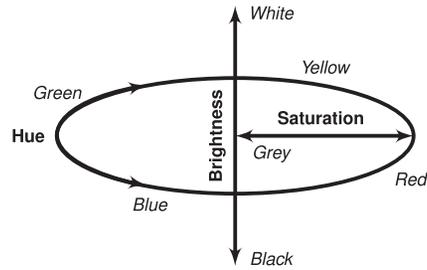


Fig. 5. The color space in terms of *hue* (radial angle in the horizontal plane), *brightness* (vertical axis), and *saturation* (horizontal radial magnitude), which are integral dimensions and therefore form a domain.

process that, by selectively combining two distinct concepts, leads to new concepts, called *blends* (Fauconnier and Turner 1998). Besold and Plaza (2015) constructed a conceptual blending engine based on generalization and amalgams, and Confalonieri et al. (2015) used argumentation to evaluate conceptual blends.

3 SPATIAL REPRESENTATIONS

In comparison to symbolic and connectionist representations, the importance of spatial representations was raised by Gärdenfors (2000) with the theory of conceptual spaces. Since before Gärdenfors' proposal, in the computing community, a spatial representation called the Vector Space Model (VSM) has been a popular tool for modeling many different domains and applications (Dubin 2004), with the topic model being a prominent example of concept creation. In this section, we introduce these three kinds of spatial representations and their relevance to concept creation.

3.1 Gärdenfors' Conceptual Spaces

Gärdenfors (2000) proposes a geometrical representation of concepts, named *conceptual spaces*. A conceptual space is formed by *quality dimensions*, which “correspond to the different ways stimuli are judged to be similar or different” (Gärdenfors 2000, p. 6). An archetypal example is a color space with the dimensions *hue*, *saturation* (or *chromaticism*), and *brightness*. Each quality dimension has a particular geometrical structure. For instance, hue is circular, whereas brightness and saturation have finite linear scales (Figure 5). It is important to note that the values on a dimension need not be numbers.

A *domain* is a set of *integral* (as opposed to *separable*) dimensions, meaning that no dimension can take a value without every other dimension in the domain also taking a value. Therefore, hue, saturation and brightness in the preceding color model form a single domain. A conceptual space is simply “a collection of one or more domains” (Gärdenfors 2000, p. 26). For example, a conceptual space of elementary colored shapes could be defined as a space comprising the preceding domain of color and a domain representing the perceptually salient features of a given set of shapes.

A *property* corresponds to a region of a domain in a conceptual space (and more specifically, a *natural* property corresponds to a *convex* region). A concept in Gärdenfors' formulation is represented in terms of its properties, normally including multiple domains. Interestingly, this means that property is a special, single-domain case of concept. For instance, the concept “red” is a region in the color space. It is also a property of anything that is red.

An *object* is a point in a space (i.e., a point in a certain region (property) of each of one or more domains). The spatial location of an object in a conceptual space allows the calculation of distance between objects, which gives rise to a natural way of representing similarities. The distance measure may be a true metric (e.g., Gärdenfors (2000) suggests that Euclidean distance is often suitable

with integral dimensions and cityblock distance with separable dimensions) or non-metric, such as a measure based on an ordinal relationship or the length of a path between vertices in a graph. When calculating distance, salience weights associated with each of the dimensions can be varied. It is the context in which a concept is used that determines which dimensions are the most prominent and hence have bigger weights.

Such spatial representations naturally afford reasoning in terms of spatial regions. Boundaries between regions are fluid, an aspect of the representation that may be usefully exploited by creative systems searching for new interpretations of familiar concepts. A further consequence of the geometrical nature of the representation is that conceptual spaces are particularly powerful in dealing with *concept learning* and *concept combination*. Learning can be modeled via supervised classification in terms of distance from prototypical centroids of regions, or via unsupervised clustering based on spatial distances. Combination can be understood in terms of intersection of spatial regions, or in terms of replacing values of one concept's regions by another for more complex cases where logical intersection fails—refer to Gärdenfors (2000, Sections 4.4, 4.5) for discussion.

Although Gärdenfors' theory has yet to be fully formalized in mathematical terms, several approaches to formalization of some aspects of it have appeared in the literature. Two approaches build on an initial formalization by Aisbett and Gibbon (2001). One, based on fuzzy set theory, is presented in detail by Rickard et al. (2007a), drawing on their previous work (Rickard 2006; Rickard et al. 2007b). The other, employing vector spaces, is presented by Raubal (2004), with subsequent related work by Schwering and Raubal (2005) and Raubal (2008). Moreover, Chella et al. (2004, 2007, 2008, 2015) have done substantial work in formalizing conceptual space theory and applying it to robotics. There is empirical evidence to support the theory as a cognitive model (Jäger 2010).

3.2 Vector Space Model

As Gärdenfors (2000) points out, an appropriate approach to computation with geometric representations is the use of VSMS: they provide algorithms and frameworks for classification, clustering, and similarity calculation, which lend themselves directly to some of the key questions in conceptual modeling. In text modeling, the use of VSMS is long established, having been introduced by Salton (1971) when building the SMART information retrieval system, taking the terms in a document collection as dimensions. Every document is represented by a vector of terms, where the value of each element is the (scaled) frequency of the corresponding term in the document. Each term in a document has a different level of importance, which can be represented by additional term weights in a document vector. A popular weighting schema is Term Frequency–Inverse Document Frequency (TF-IDF) (Spärck Jones 1972), which is based on the idea that terms that appear in many documents are less important for a single document. To see how similar two documents are, their vectors can be compared, with the most commonly used similarity measure being the cosine value of the angle between the document vectors (Salton 1989). Such a VSM assumes pairwise orthogonality between term vectors (the columns), which generally does not hold due to correlation between terms. The Generalized Vector Space Model (GVSM) provides a solution to this problem (Wong et al. 1985; Raghavan and Wong 1986).

However, the same matrix can be used to calculate the similarity between two *terms* (words) by taking a different perspective: a term can be represented by a vector over the documents where they appear (i.e., if document vectors are rows in the document-term matrix, term vectors are the columns). This approach has been exploited by Turney and Pantel (2010) to build models of lexical meaning that reflect human similarity judgments by reducing the co-occurrence context from the scale of whole documents to windows of a few words. In a contrasting approach, word vectors with similar properties are learned by neural networks (e.g., see Mikolov et al. (2013)), and these are good at capturing syntactic and semantic regularities, often remaining computationally efficient and

retaining low dimensionality—refer to Baroni et al. (2014) for a review and comparison. A range of distance measures can also be used with these models; although cosine distance is the most commonly used, others can be more suited to different domains and tasks, and Kiela and Clark (2014) show a mean-weighted cosine distance variant to be most accurate in reflecting human judgments.

Extending the preceding VSMS to model the compositional meaning of phrases and sentences (rather than individual words) is the subject of much current research, with a range of methods including hierarchical compression using neural auto-encoders (e.g., Socher et al. (2013)), sequence modeling using convolutional networks (e.g., Kalchbrenner et al. (2014)), and categorical combination using tensor operations (e.g., Coecke et al. (2011)). Extension beyond the sentence to models of discourse meaning is also being investigated (e.g., Kalchbrenner and Blunsom (2013)).

In addition to word-context matrices, pair-pattern matrices have been used in VSMS, where rows correspond to pairs of terms and columns correspond to the patterns in which the pairs occur. They are used to measure the similarity of semantic relations in word pairs (Lin and Pantel 2001; Turney and Littman 2003). Higher-order tensors (matrices are second-order tensors), such as a word-word-pattern tensor, have also been found to be useful in measuring the similarity of words (Turney 2007).

VSM-based models have been used in generating spoken dialogues (Wen et al. 2016) and modern haikus (Wong and Chun 2008). Venour et al. (2010) constructed a novel semantic space, where the distance between words reflects the difference in their styles or tones, as part of generating linguistic humors. Juršič et al. (2012) took advantage of a document-term matrix and centroid vectors to find bridging terms of two literatures. In addition to constructing VSMS from text, vectors of RGB color values were used by de Melo and Gratch (2010) to evolve emotional expressions of virtual humans. Thorogood and Pasquier (2013) used vectors of low-level audio features in generating audio metaphors. Maher et al. (2013) used vectors of attributes (e.g., display area, amount of memory, and CPU speed) to measure *surprise*. Furthermore, the future applications of VSMS in computational creativity tasks were discussed by McGregor et al. (2014).

3.3 Topic Model

Topic modeling is a general approach to modeling text using VSMS. It assumes that documents (or pieces of text) are composed of, or generated from, some underlying *latent concepts* or *topics*. One of the earliest variants, Latent Semantic Analysis (LSA) (Landauer et al. 1998) has become a widely used technique for measuring similarity of words and text passages. LSA applies singular value decomposition (SVD) to a standard word-document (or word-context, where *context* refers to some window of words around a term) matrix; the resulting eigenvectors can be seen as *latent concepts*, as they provide a set of vectors that characterize the data but abstract away from the surface words while relating words used in similar contexts to each other. By using these concept vectors as the bases, we obtain a new *latent semantic space*, and by limiting them to the eigenvectors with the largest values, this space can have a drastically smaller number of dimensions while still closely approximating the original. LSA is not limited to words and their contexts. It can be generalized to unitary event types and the contexts in which instances of the event types appear (e.g., bag-of-features in computer vision problems (Sivic et al. 2005)); it has been successfully applied in many tasks, including topic segmentation (Olney and Cai 2005). However, although the number of dimensions chosen for this latent semantic space is critical for performance, there is no principled way of doing it. Moreover, the dimensions in the new space do not have obvious interpretations.

An approach that solves some of these problems is Probabilistic Latent Semantic Indexing (PLSI) (Hofmann 1999). PLSI can be seen as a probabilistic variant of LSA; rather than applying SVD to derive latent vectors by factorization, it fits a statistical latent class model on a word-context matrix

using Tempered Expectation Maximization (TEM). This process still generates a low-dimensional latent semantic space in which dimensions are topics, but now these topics are probability distributions over words (i.e., sets of words with a varied degree of membership to the topic, which we can see as latent concepts here), and documents are probabilistic mixtures of these topics. The number of dimensions in the new space is determined according to the statistical theory for model selection and complexity control. This can be used directly to model document content and similarity, or for example, embedded within an aspect Markov model to segment and track topics (Blei and Moreno 2001).

A shortcoming of PLSI, however, is that it lacks a generative model of document-topic probabilities: it therefore must estimate these from topic-segmented training data and is not directly suitable for assigning probability to a previously unseen document, instead requiring an additional estimation process during decoding (Blei and Moreno 2001). These are addressed by models such as Latent Dirichlet Allocation (LDA) (Blei et al. 2003), which take a similar latent-variable approach but make it fully Bayesian, allowing topics to be inferred without prior knowledge of their distribution. LDA has been used very successfully for fully unsupervised induction of topics from text in many domains (e.g., see Griffiths et al. (2005) and Hong and Davison (2010)). It requires the number of topics and certain hyper-parameters to be specified; but even these can be estimated by hierarchical Bayesian variants (e.g., see Blei et al. (2004)).

Variants of LDA that incorporate data from outside the text can then go even further toward full concept discovery by inducing topics with relations between text and author, social network properties, and so on (refer to Blei (2012) for an overview). This has been particularly important in social media modeling, where texts themselves are very short. Here, extended variants have been developed and successfully applied in many ways, with the notion of topic (or concept) depending on objective: for example, to discover and model health-related topics (Paul and Dredze 2014), to model topic-specific influences by incorporating information about network structures (Weng et al. 2010), to detect and profile breadth of interest in audiences using timeline-aggregated data (Concannon and Purver 2014), to build predictive models for review sites by including user- and location-specific information (Lu et al. 2016), and to help predict stock market movements via incorporating sentiment aspects (Nguyen and Shirai 2015).

Topic models have been used in various ways in the computational creativity community. Straparava et al. (2007) used LSA to compute *lexical affective semantic similarity* to generate animated advertising messages. Topic vectors are used for conceptual blending by Veale (2012). Xiao and Blat (2013) used an LSA space built from Wikipedia to generate pictorial metaphors.

4 CONNECTIONIST REPRESENTATIONS

Connectionist representations are composed of interconnected simple units, featuring parallel distributed processing. Hebb (1949) proposes that concepts are represented in the brain in terms of neural assemblies. The neural blackboard architecture (van der Velde and de Kamps 2006) suggests a way of combining neural assemblies to account for higher-level human cognition. The most commonly used family of connectionist models is artificial neural networks (ANNs). A more complex version of ANNs, deep neural networks, provides representations at a series of abstraction levels. In this section, we introduce these four conceptual representations and their relevance to concept creation.

Given their “network-alike” look, these connectionist representations may resemble semantic networks (Section 2.3), ontologies (Section 2.4), and information networks (Section 2.5). However, in each of the conceptual representations, the relations (and the way of interaction) between the units of a “network” are fundamentally different. In particular, most connectionist representations are procedural. They have no explicit representations of concepts: the representations are rather

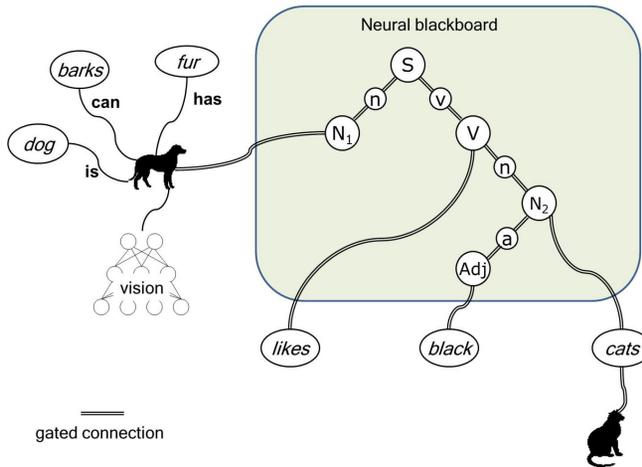


Fig. 6. Neural assembly for dog, embedded in a relation structure based on a neural blackboard architecture.

distributed in the network and made explicit only when the connectionist representation is executed. For more detailed discussions on ANN frameworks for distributed representations, refer to Kröse and van der Smagt (1993) and Hinton et al. (1986).

4.1 Neural Assembly and Neural Blackboard

Hebb (1949) proposes that concepts are represented in the brain in terms of neural assemblies. In other words, in modern terminology, one can say that Hebb spoke about concepts in the brain (e.g., Abeles (2011)). This is also in agreement with Hebb’s hypothesis that thinking consists of sequential activations of assemblies, or *phase sequences*, as he called it (e.g., refer to Harris (2005) for a recent analysis). A neural assembly is a group of neurons that are strongly interconnected. As a consequence, when a part of an assembly is activated (e.g., by perception), it can reactivate the other parts of the assembly as well. For example, when we see an animal, certain neurons in the visual cortex will be active. But when the animal makes a sound, certain neurons in the auditory cortex will be active as well. Neural assemblies arise over time, based on learning processes such as Hebbian learning (Hebb 1949) (i.e., neurons that fire together wire together). Over time, other neurons could become a part of the assembly as well, particularly when they are consistently active together with the assembly. Examples are the neurons that represent the word we use to name the animal, or neurons involved in our actions or emotions when we encounter it (van der Velde 2015).

Figure 6 illustrates (parts of) a neural assembly that could represent the concept “dog.” It would consist of the neurons involved in perceiving the animal, as well as neurons representing the relations *is dog*, *can bark*, or *has fur*, and neurons that represent other aspects of our (e.g., emotional) experiences with dogs. Figure 6 does not imply that the ovals representing concepts are semantically meaningful on their own. Each “concept node” in the figure represents a neural assembly, consisting of a network structure that integrates all aspects of the concept. In particular, it represents the interconnection of perception and action components of the concept, which represents the grounding of the concept in behavior, and thus (part of) its semantics (e.g., see van der Velde (2015)).

A fundamental issue with neural assemblies is how they can account for the non-associative aspects of human cognition (van der Velde 1993). Of course, associations are crucial, because without them we could not survive in any given environment. But for high-level cognition (e.g., language, reasoning), associations are not enough. Instead, relations are necessary, because they provide the

basis for systematic knowledge. For example, we can apply the relation *greater than* to any pair of objects, not just the ones we happen to be acquainted with. In contrast, associations are always coincidental. For example, in the classical Pavlov experiment, the sound of the bell was associated with the food, but it could have been any other stimulus (as was indeed tested by Pavlov). Thus, relational knowledge cannot be established on the basis of associations alone.

The fact that in human high-level cognition relations are implemented in neural networks can result in a mixed representation, in which relations and associations are combined. For example, frequently occurring relations (e.g., *dogs chase cats*) could result in a more directly associative link between the concepts involved (“dogs” and “cats”). Such more direct associations are sometimes used in idioms like “They fight like cats and dogs.”

As noted, associations are direct links (connections) between neural entities (e.g., neurons or neural populations). Associative links, when they are available, result in very fast activations. In contrast, the more elaborate links provided by relations, which also require the specific type of relation to be expressed, operate more slowly. In this way, forming associations on top of relations can provide for a double response; a fast one based on the associations and a slower one based on the relations. This combination can help in, say, hazardous circumstances, where speed is essential. It could also reduce the processing time in specific cases. However, on the reverse side, it could also lead one astray from the correct analysis.

Van der Velde and de Kamps (2006) present a neural architecture in which (temporal or more permanent) connections between conceptual representations based on neural assemblies could be formed. Figure 6 illustrates the relation *dog likes black cats*, where the neural assemblies representing the concepts “dog,” “likes,” “black,” and “cats” are interconnected in a *neural blackboard*. The blackboard consists of neural populations that represent the specific types of concepts and the relations between them. Thus, N_1 and N_2 represent a noun, V a verb, Adj an adjective, and S a clause. The connections in the blackboard are gated connections (consisting of neural circuits). Gated connections provide control of activation, which allows the representation of relations and hierarchical structures, as found in higher-level human cognition. In this way, a temporal connection can be formed between the conceptual assemblies and populations of the same type in the blackboard (“dog” with N_1 , “cats” with N_2 , “like” with V , “black” with Adj) and between the type populations themselves, which results in the representation for the clause (relation) *dog likes black cats*.

4.2 Artificial Neural Network

ANNs draw inspiration from biological neurons, although their usage is usually not to model human brains (Sun 2008). The building blocks of ANNs are simple computational units that are highly interconnected. The connections between the units determine the function of a network. In ANNs, concepts are implicitly represented by four parts: the network-wide algorithm, the network topology, the computational procedures in each individual unit, and the weights of their connections. Executing such algorithms produces explicit representations of concepts in the form of activation patterns, although individual nodes (e.g., in the output layer of an ANN classifier) can also be seen as outward-facing representations of concepts (e.g., the activation of an output node is regarded as a representation of the corresponding class/concept). It is, however, important to highlight that most individual nodes of neural networks carry no recognizable semantic value.

ANNs can be viewed as directed weighted graphs in which artificial neurons are nodes and edges are connections between neuron outputs and neuron inputs. Based on the connection patterns (architecture), ANNs can be grouped into two categories: *feed-forward neural networks*, in which graphs have no loops, and *recurrent (or feedback) neural networks* (Medsker and Jain 1999), in which loops occur due to feedback connections. The most common family of feed-forward

neural networks is *multilayer perceptron* (Haykin 1994). Some of the well-known recurrent neural networks are the Elman Network (Cruse 1996), the Hopfield Network (Gurney 1997), and the Boltzmann Machine (Hinton and Salakhutdinov 2006).

After the learning phase, standard feed-forward networks usually produce only one set of output values, rather than a sequence of values, for a given input. They are also memory-less in the sense that their responses to inputs are independent of the previous network states. Feed-forward neural networks are usually used as classifiers by learning non-linear relationships between inputs and outputs. Typically, the nodes in the output layer of a feed-forward ANN correspond to regions in the input feature space and thus can be seen as representing centroids, or prototypes, of such conceptual regions. Recurrent, or feedback, neural networks, however, are dynamic systems. Because of the feedback paths, the inputs to neurons are then modified, which leads the network to enter a new state. Regarding the representation of temporal concepts, recurrent neural networks can be trained to learn and predict each successive symbol of any sequence in a particular language.

ANNs learn by iteratively updating the connection weights in a network, toward better performance on a certain specific task. There are three main learning paradigms: *supervised*, *unsupervised*, and *hybrid*. Considering that ANNs can learn patterns of neuron activation (both simultaneous and sequential activations), they can be used to simulate creative processes, such as via combining two or more patterns into a single one, or creating a random variation of a learned pattern.

Various types of ANNs have been used in the music domain for melody generation (Todd 1992; Eck and Schmidhuber 2002; Hoover et al. 2012) and improvisation (Bown and Lexer 2006; Smith and Garnett 2012). ANNs are also suitable for evaluation, as for music (Tokui and Iba 2000; Monteith et al. 2010), images (Baluja et al. 1994; Machado et al. 2008; Norton et al. 2010), poetry (Levy 2001), and culinary recipes (Morris et al. 2012). Saunders and Gero (2001) used a particular type of ANN, the self-organizing map (SOM), to evaluate the novelty of images newly encountered by an agent. An ANN was also used by Bhatia and Chalup (2013) to measure surprise. In addition, Terai and Nakagawa (2009) used a recurrent neural network to generate metaphors.

4.3 Deep Neural Network

Deep networks are a recent extension of the family of connectionist representations, which attempt to model high-level abstractions of data using *deep architectures* (Bengio et al. 2013). Deep architectures are composed of multiple levels of non-linear operations, such as neural nets with many hidden layers. This usually results in a stack of “local” networks whose types need not be the same across the entire deep representation.

The human brain is organized in a deep architecture (Serre et al. 2007). An input percept is represented at multiple levels of abstraction, with each level corresponding to a different area of the cortex. The brain also appears to process information through multiple stages of transformation. This is particularly evident in the primate visual system, with its sequence of processing stages, from detecting edges to primitive shapes to gradually more complex shapes.

Deep representations are built with *deep learning* techniques (Bengio et al. 2013; LeCun et al. 2015). Deep learning algorithms typically operate on networks with fixed topology and solely adjust the weights of the connections. Each type of deep architectures is amenable to specific learning algorithms: for example, *deep convolutional networks* are usually trained with backpropagation (e.g., see Kalchbrenner et al. (2014)), whereas *deep belief networks* (Hinton et al. 2006; Hinton 2009) are obtained by stacking several Restricted Boltzmann Machines (Hinton and Salakhutdinov 2006), each of which is typically trained with the contrastive divergence algorithm. Deep belief networks are based on probabilistic approaches, whereas other approaches exist, such as *auto-encoders*, which are based on reconstruction-based algorithms, and *manifold learning*, which has

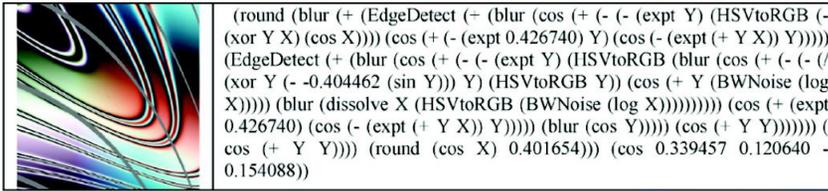


Fig. 7. An example phenotype (image at left) and the expression-based genotype (code that produced the image at right) (Machado et al. 2007). © Machado et al. Reproduced by permission.

roots in geometrical approaches. Although the stacked layers may allow the network to effectively learn the intricacies of the input, the fact that they usually have a fixed topology imposes representation and learning limits *a priori*. In contrast, a deep learning algorithm for dynamic topologies, allowing the creation of new nodes or layers of nodes, would enable the creation of new concepts and new dimensions in a conceptual space.

Deep representations and learning have been used in modeling and generating language (Section 3.2), producing jazz melodies (Bickerman et al. 2010), creating spaceships for 2D arcade-style computer games (Liapis et al. 2013), generating images (Goodfellow et al. 2014; Gregor et al. 2015), transferring visual styles (Gatys et al. 2015), and as part of a computational framework of imagination (Heath et al. 2015).

5 PROCEDURAL REPRESENTATIONS

A procedural representation of an artifact specifies a procedure (e.g., a program) that once executed produces the artifact being represented. To illustrate the difference between descriptive and procedural representations, we will resort to an example in the musical domain, the task of representing a sequence of pitches. Using a *descriptive* approach, one could, for instance, use a vector of pitch values to represent this sequence. A *procedural* representation would, for instance, use a program to generate the sequence of pitches by performing a set of operations on sub-sequences. An example is the GP-Music system (Johanson and Poli 1998):

“The item returned by the program tree in the GP-Music System is not a simple value but a note sequence. Each node in the tree propagates up a musical note string, which is then modified by the next higher node. In this way a complete sequence of notes is built up, and the final string is returned by the root node. Note also that there is no input to the program; the tree itself specifies a complete musical sequence.”

It is straightforward to conceive a naive descriptive representation, because we can always resort to the enumeration of the elements of the artifact. Therefore, one should ponder about the motivation for using procedural representations. A program can take advantage of the structure of an artifact—such as repetition of note sequences, relations between sequences, and cycles—and as such, the size of the procedural representation of an artifact that has structure can be significantly smaller than the size of its descriptive representation. Additionally, it is also easier to induce structural changes, in the case of creating new concepts.

Procedural representations are particularly popular for image creation in Evolutionary Computation (EC). Many of them are expression based, such as the example in Figure 7 showing both a symbolic expression and the corresponding image produced by this expression. Some notable examples of expression-based EC are by Sims (1991), Rooke (1996), Unemi (1999), Saunders and Gero (2001), Machado and Cardoso (2002), and Hart (2007).

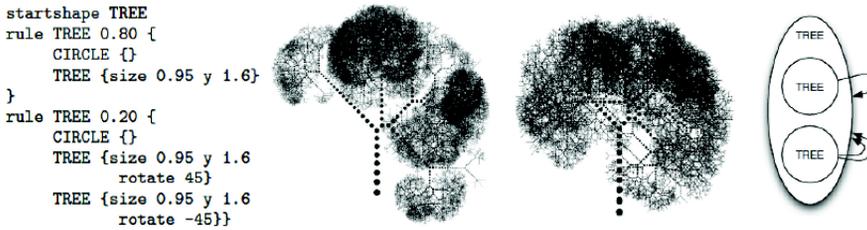


Fig. 8. Examples of non-deterministic grammar, their corresponding tree-like shapes, and their graph representation (Machado et al. 2010). © Machado et al. (2010) Reproduced by permission.

Machado et al. (2010) evolved non-deterministic context-free grammars. The grammars are represented by means of a hierarchic graph, which is manipulated by graph-based crossover and mutation operators. The context-free grammar constitutes a set of program instructions that are executed to generate the visual artifacts; thus, although the grammar has a symbolic representation, the representation of the image is procedural. One of the novel aspects of this approach is that each grammar has the potential to represent, and generate, a family of akin shapes (Figure 8).

Zhu and Mumford (2007) used stochastic context-sensitive grammars embedded in an And-Or graph to represent large-scale visual knowledge, using raster images as input, for modeling and learning. In their preliminary works, they show that the grammars enable them to parse images and construct descriptive models of images. This allows the production of alternative artifacts and the learning of new models.

Byrne et al. (2012) evolved architectural models using *grammatical evolution*. Grammatical evolution is a grammar-based form of Genetic Programming (GP), replacing the parse-tree based structure of GP with a linear genome. It generates programs by evolving an integer string to select rules from a user-defined grammar. The rule selections build a derivation tree that represents a program. Any mutation or crossover operators are applied to the linear genome instead of the tree itself. McDermott (2013) also used grammatical evolution to evolve graph grammars in the context of evolutionary 3D design. Greenfield (2012) used GP to evolve controllers for drawing robots. The author resorted to an assembly language where each statement is represented as a triple. The programs assume the form of a tree.

Music, or more specifically composition as a creative process, has been another common application for procedural representations. One of the first, if not the first, evolutionary approaches to music composition resorts to a procedural representation. Horner and Goldberg (1991) used the Genetic Algorithm (GA) for evolving sequences of operations that transform an initial note sequence into a final desired sequence within a certain number of steps. Putnam (1994) was one of the first to use GP for music generation purposes. He used the traditional GP tree structures to interactively evolve sounds. Spector and Alpern (1994) used GP to evolve programs that transform an input melody by applying several transformation functions (e.g., invert, augment, and transpose). The work of Johanson and Poli (1998) constitutes another early application of GP in the music domain. Hornel and Ragg (1996) evolved the weights of neural networks that perform harmonization of melodies in different musical styles. McCormack (1996) explored stochastic methods for music composition and proposed evolving the transition probability matrix for Markov models. Monmarché et al. (2007) used artificial ants to build a melody according to transition probabilities while also taking advantage of the collective behavior of the ants marking paths with pheromones. They evolved graph-like structures, the vertices are MIDI events, and a melody corresponds to a path through several vertices. McCormack (1996) focused on grammar-based approaches for music composition, exploring the use of *L-systems*. In an earlier work, McCormack (1993) used L-systems for evolving 3D shapes.

Table 2. Concept Creation Methods Applied at Different Levels and Types of Conceptual Representations

Level/Type of Representation	Concept Creation Method			
	Extraction	Induction	Recycling	Exploration
Symbolic level	✓	✓	✓	✓
Spatial level		✓		✓
Connectionist level		✓	✓	✓
Descriptive	✓	✓	✓	✓
Procedural	✓	✓	✓	✓

6 CONCLUSION

We have structured this review according to three levels of representation (symbolic, spatial, connectionist), inspired by Gärdenfors (2000), and separately considered additional procedural representations. In an Online Supplement, we additionally reviewed domain-specific representations in popular application areas of computational creativity. It is our hope that this organization will act as a map, helping researchers navigate in the forest of conceptual representations used in computational concept creation.

In Section 1, we also gave a taxonomy of concept creation approaches, where the main classes of methods are concept extraction, concept induction, concept recycling, and concept space exploration. We next summarized the results of this review by considering relations between the different levels of conceptual representations, the application domains, and the types of concept creation methods.

Consider the application domains first. In the Online Supplement, we considered representations in four major application domains of concept creation—language, music, image, and emotion. In addition to the domain-specific concept representations, the generic representations of Sections 2 through 5 can also be used in these domains. Domain-generic representations are especially useful in applications that process information from multiple domains. It actually turns out that in all of the domains that we have considered, conceptual representations have been used from all of the levels and categories used to structure this review (symbolic, spatial, connectionist; declarative, procedural).

Take for instance text documents. At the symbolic level, a document as a sequence of words is ready for people to read. At the spatial level, documents are routinely represented as vectors, allowing, for example, comparison of semantic similarity between documents. At the connectionist level, text can be generated by activating an ANN that captures characteristics of a collection of documents. The connectionist representation is a procedural one, whereas the spatial and symbolic representations used in the example are declarative ones.

In Table 2, we summarize, based on the preceding review, how concept creation methods relate to the different levels and types of conceptual representations. Two interesting observations can be made. First, concept extraction has mainly been applied at one of the three levels only—the symbolic level—between different symbolic representations. This is for an obvious reason: symbolic representations are often designed to be manipulated and translated, and at the very least, by definition, have meanings that can be processed as symbols. Spatial and especially connectionist representations lend them much less for such translation, if at all. According to Gärdenfors (2000), the three levels are connected so that the connectionist level feeds spatial representations, which in turn become symbolic in language. In the same way, it seems plausible that concept extraction methods operating between levels could be developed. For instance, McGregor et al. (2015) take

steps toward establishing such a mapping between spatial and symbolic levels. The second observation is that concept induction, concept recycling, and concept space exploration have all been used for almost all of the levels and types of conceptual representations, with the exception that we are not aware of applications of concept recycling to spatial representations. This seems to be a promising area for research in concept creation: spatial representations lend themselves for mutation and combination, and the question is more in how to utilize this capability in concept creation in a useful way.

This review demonstrates that conceptual representations at each of the symbolic, spatial, and connectionist levels, as well as both descriptive and procedural representations, are abundant. Still, promising new representations are emerging at all levels, such as bisociation (Section 2.1), heterogeneous information network (Section 2.5), conceptual spaces (Section 3.1), neural blackboard (Section 4.1), and deep neural network (Section 4.3).

Furthermore, numerous avenues exist for research into computational concept creation and conceptual representations, and we mention some of them here. For instance, in the field of concept extraction, an interesting possibility for future work is automatically building plan operators. Concerning concept induction, a particularly interesting line of future work, is learning new nomenclological categories. In terms of concept recycling, the combination of thematically different ontologies can be a new approach for dealing with analogies, metaphors, pataphors, and conceptual blending. Regarding concept space exploration, an interesting opportunity is discovering novel concepts in word association graphs by exploiting graph structure measures. Finally, with respect to transformational creativity, finding new interpretations of familiar concepts in a conceptual space model would offer ways to be creative beyond the original limits.

REFERENCES

- Agnar Aamodt and Enric Plaza. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications* 7, 1 (1994), 39–59.
- Moshe Abeles. 2011. Cell assemblies. *Scholarpedia* 6, 7 (2011), 1505.
- Lada A. Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social Networks* 25, 3 (2003), 211–230.
- Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM Conference on Digital Libraries (DL'00)*. 85–94.
- Janet Aisbett and Greg Gibbon. 2001. A general formulation of conceptual spaces as a meso level representation. *Artificial Intelligence* 133, 1–2 (2001), 189–232.
- Ahmed F. AlEroud and George Karabatis. 2013. A system for cyber attack detection using contextual semantics. In *Proceedings of the 7th International Conference on Knowledge Management in Organizations: Service and Cloud Computing*. 431–442.
- Shumeet Baluja, Dean Pomerleau, and Todd Jochem. 1994. Towards automated artificial evolution for computer-generated images. *Connection Science* 6 (1994), 325–354.
- Albert-Laszlo Barabási, Hawoong Jeong, Zoltan Nédá, Erzsebet Ravasz, Andras Schubert, and Tamas Vicsek. 2002. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and Its Applications* 311, 3 (2002), 590–614.
- Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. 238–247.
- Atilim Günes Baydin, Ramon López de Mántaras, and Santiago Ontañón. 2012. Automated generation of cross-domain analogies via evolutionary computation. In *Proceedings of the 3rd International Conference on Computational Creativity (ICCC'12)*. 25–32.
- Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828.
- Matthew Berland and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*. 57–64.
- Michael R. Berthold. 2012. Towards bisociative knowledge discovery. In *Bisociative Knowledge Discovery: An Introduction to Concept, Algorithms, Tools, and Applications*, M. R. Berthold (Ed.). Springer, Berlin, Germany, 1–10.

- Tarek R. Besold and Enric Plaza. 2015. Generalize and blend: Concept blending based on generalization, analogy, and amalgams. In *Proceedings of the 6th International Conference on Computational Creativity (ICCC'15)*.
- Shashank Bhatia and Stephan Chalup. 2013. A model of heteroassociative memory: Deciphering surprising features and locations. In *Proceedings of the 4th International Conference on Computational Creativity (ICCC'13)*. 139–146.
- Greg Bickerman, Sam Bosley, Peter Swire, and Robert M. Keller. 2010. Learning to create jazz melodies using deep belief nets. In *Proceedings of the 1st International Conference on Computational Creativity (ICCC'10)*. 228–237.
- Chris Biemann. 2005. Ontology learning from text: A survey of methods. *LDV Forum* 20, 2 (2005), 75–93.
- Eduardo Blanco, Nuria Castell, and Dan Moldovan. 2008. Causal relation extraction. In *Proceedings of the 6th International Language Resources and Evaluation (LREC'08)*. 310–313.
- David Blei. 2012. Probabilistic topic models. *Communications of the ACM* 55, 4 (2012), 77–84.
- David Blei, Thomas Griffiths, Michael Jordan, and Joshua Tenenbaum. 2004. Hierarchical topic models and the nested Chinese restaurant process. *Advances in Neural Information Processing Systems* 16 (2004), 17–24.
- David Blei and Pedro Moreno. 2001. Topic segmentation with an aspect hidden Markov model. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 343–348.
- David Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Machine Learning Research* 3 (2003), 993–1022.
- Margaret A. Boden. 1990. *The Creative Mind: Myths and Mechanisms*. Basic Books, New York, NY.
- Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. 2015. Semeval-2015 task 17: Taxonomy extraction evaluation. In *Proceedings of the 9th International Workshop on Semantic Evaluation (TextEval'15)*.
- Maroua Bouzid, Carlo Combi, Michael Fisher, and Gérard Ligozat. 2006. Guest editorial: Temporal representation and reasoning. *Annals of Mathematics and Artificial Intelligence* 46, 3 (2006), 231–234.
- Oliver Bown and Sebastian Lexer. 2006. Continuous-time recurrent neural networks for generative and interactive musical performance. In *Applications of Evolutionary Computing*. Springer, 652–663.
- Ronald J. Brachman and Hector J. Levesque (Eds.). 1985. *Readings in Knowledge Representation*. Morgan Kaufmann, San Francisco, CA.
- Sergey Brin. 1999. Extracting patterns and relations from the World Wide Web. In *Selected Papers From the International Workshop on the World Wide Web and Databases (WebDB'98)*. 172–183. <http://dl.acm.org/citation.cfm?id=646543.696220>
- Peter Bruza and Marc Weeber. 2008. *Literature-Based Discovery*. Germany.
- Jonathan Byrne, Erik Hemberg, Anthony Brabazon, and Michael O'Neill. 2012. A local search interface for interactive evolutionary architectural design. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, P. Machado, J. Romero, and A. Carballal (Eds.). Lecture Notes in Computer Science, Vol. 7247. Springer, 23–34.
- Andrea Capocci, Vito D. P. Servedio, Guido Caldarelli, and Francesca Colaiori. 2005. Detecting communities in large networks. *Physica A: Statistical Mechanics and Its Applications* 352, 2 (2005), 669–676.
- Sharon A. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*. 120–126.
- Antonio Chella. 2015. A cognitive architecture for music perception exploiting conceptual spaces. In *Applications of Conceptual Spaces: The Case for Geometric Knowledge Representation*. Springer, 187–203.
- Antonio Chella, Silvia Coradeschi, Marcello Frixione, and Alessandro Saffiotti. 2004. Perceptual anchoring via conceptual spaces. In *Proceedings of the AAAI-04 Workshop on Anchoring Symbols to Sensor Data*.
- Antonio Chella, Haris Dindo, and Ignazio Infantino. 2007. Imitation learning and anchoring through conceptual spaces. *Applied Artificial Intelligence* 21, 4 (2007), 343–359.
- Antonio Chella, Marcello Frixione, and Salvatore Gaglio. 2008. A cognitive architecture for robot self-consciousness. *Artificial Intelligence in Medicine* 44, 2 (2008), 147–154.
- Massimiliano Ciaramita, Aldo Gangemi, Esther Ratsch, Jasmin Šaric, and Isabel Rojas. 2005. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*. 659–664.
- Philipp Cimiano, Andreas Hotho, and Steffen Staab. 2005. Learning concept hierarchies from text corpora using formal concept analysis. *Artificial Intelligence Research* 24, 1 (2005), 305–339.
- Bob Coecke, Mehrnoosh Sadzadeh, and Stephen Clark. 2011. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis* 36, 1–4 (2011), 345–384.
- Simon Colton and Geraint A. Wiggins. 2012. Computational creativity: The final frontier? In *Proceedings of the 20th Biennial European Conference on Artificial Intelligence (ECAI'12)*. 21–26.
- Shauna Concannon and Matthew Purver. 2014. Inferring cultural preference of arts audiences through Twitter data. In *Proceedings of the Conference on Digital Intelligence*.
- Roberto Confalonieri, Joe Corneli, Alison Pease, Enric Plaza, and Marco Schorlemmer. 2015. Using argumentation to evaluate concept blends in combinatorial creativity. In *Proceedings of the 6th International Conference on Computational Creativity (ICCC'15)*.

- Holk Cruse. 1996. *Neural Networks as Cybernetic Systems*. Thieme, Stuttgart, Germany.
- Amitava Das and Björn Gambäck. 2014. Poetic machine: Computational creativity for automatic poetry generation in bengali. In *Proceedings of the 5th International Conference on Computational Creativity (ICCC'14)*.
- Dmitry Davidov and Ari Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL-44)*. 297–304.
- Darcy Davis, Ryan Lichtenwalter, and Nitesh V. Chawla. 2011. Multi-relational link prediction in heterogeneous information networks. In *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining (ASONAM'11)*. 281–288.
- Randall Davis, Howard E. Shrobe, and Peter Szolovits. 1993. What is a knowledge representation? *AI Magazine* 14, 1 (1993), 17–33.
- Celso M. de Melo and Jonathan Gratch. 2010. Evolving expression of emotions through color in virtual humans using genetic algorithms. In *Proceedings of the 1st International Conference on Computational Creativity (ICCC'10)*. 248–257.
- Luciano Del Corro and Rainer Gemulla. 2013. ClausIE: Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*. 355–366.
- Luca Donetti and Miguel A. Munoz. 2004. Detecting network communities: A new systematic and efficient algorithm. *Journal of Statistical Mechanics: Theory and Experiment* 2004, 10 (2004), P10012.
- David Dubin. 2004. The most influential paper Gerard Salton never wrote. *Library Trends* 52, 4 (2004), 748–764.
- Werner Dubitzky, Tobias Kötter, Oliver Schmidt, and Michael R. Berthold. 2012. Towards creative information exploration based on Koestler's concept of bisociation. In *Bisociative Knowledge Discovery: An Introduction to Concept, Algorithms, Tools, and Applications*, M. R. Berthold (Ed.). Springer, 11–32.
- Douglas Eck and Juergen Schmidhuber. 2002. Learning the long-term structure of the blues. In *Proceedings of the 16th International Conference on Artificial Neural Networks (ICANN'02)*. 284–289.
- Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open information extraction: The second generation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence—Volume One (IJCAI'11)*. 3–10.
- Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*. 1535–1545.
- Gilles Fauconnier and Mark Turner. 1998. Conceptual integration networks. *Cognitive Science* 22, 2 (1998), 133–187.
- Christiane Fellbaum (Ed.). 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. MIT Press, Cambridge, MA.
- Jerry A. Fodor. 1975. *The Language of Thought*. Harvard University Press, Cambridge, MA.
- Blaž Fortuna, Dunja Mladenić, and Marko Grobelnik. 2006. Semi-automatic construction of topic ontologies. In *Semantics, Web and Mining*, M. Ackermann, B. Berendt, M. Grobelnik, A. Hotho, D. Mladenić, G. Semeraro, et al. (Eds.). Lecture Notes in Computer Science, Vol. 4289. Springer, 121–131.
- Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. 2002. Sweetening ontologies with DOLCE. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web (EKAW'02)*. 166–181.
- Peter Gärdenfors. 2000. *Conceptual Spaces: The Geometry of Thought*. MIT Press, Cambridge, MA.
- Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. arXiv:1508.06576.
- Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2006. Automatic discovery of part-whole relations. *Computational Linguistics* 32, 1 (March 2006), 83–135.
- Roxana Girju and Dan I. Moldovan. 2002. Text mining for causal relations. In *Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference*. 360–364.
- Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 task 04: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval'07)*. 13–18.
- Robert L. Goldstone and Andrew T. Hendrickson. 2010. Categorical perception. *Wiley Interdisciplinary Reviews: Cognitive Science* 1, 1 (2010), 69–78.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, et al. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27* (2014), 2672–2680.
- Gérard Govaert and Mohamed Nadif. 2014. *Co-Clustering: Models, Algorithms and Applications*. Wiley, London, UK.
- Gary Greenfield. 2012. A platform for evolving controllers for simulated drawing robots. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, P. Machado, J. Romero, and A. Carballal (Eds.). Lecture Notes in Computer Science, Vol. 7247. Springer, 108–116.
- Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. 2015. DRAW: A recurrent neural network for image generation. arXiv:1502.04623.

- Thomas L. Griffiths, Mark Steyvers, David Blei, and Joshua B. Tenenbaum. 2005. Integrating topics and syntax. In *Advances in Neural Information Processing Systems* 17 (2005), 537–544.
- Tom Gruber. 2009. Ontology. In *Encyclopedia of Database Systems*, T. Özsu and L. Liu (Eds.). Springer, Boston, MA, 1963–1965.
- Miha Grčar, Nejc Trdin, and Nada Lavrač. 2013. A methodology for mining document-enriched heterogeneous information networks. *Computer* 56, 3 (2013), 321–335.
- Nicola Guarino and Pierdaniele Giaretta. 1995. Ontologies and knowledge bases: Towards a terminological clarification. In *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*. IOS Press, Amsterdam, Netherlands, 25–32.
- Kevin Gurney. 1997. *An Introduction to Neural Networks*. CRC Press, Boca Raton, FL.
- Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena* 42, 1 (1990), 335–346.
- Kenneth D. Harris. 2005. Neural signatures of cell assembly organization. *Nature Reviews Neuroscience* 6 (2005), 399–407.
- David A. Hart. 2007. Toward greater artistic control for interactive evolution of images and animation. In *Proceedings of the 2007 EvoWorkshops on EvoCoMnet, EvoFIN, EvoIASP, EvoINTERACTION, EvoMUSART, EvoSTOC and EvoTransLog*. 527–536.
- John Haugeland. 1985. *Artificial Intelligence: The Very Idea*. MIT Press, Cambridge, MA.
- Simon S. Haykin. 1994. *Neural Networks: A Comprehensive Foundation*. Prentice Hall.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics (COLING'92)*. 539–545.
- Derrall Heath, Aaron Dennis, and Dan Ventura. 2015. Imagining imagination: A computational framework using associative memory models and vector space models. In *Proceedings of the 6th International Conference on Computational Creativity (ICCC'15)*.
- Donald O. Hebb. 1949. *The Organization of Behavior: A Neuropsychological Theory*. Wiley, New York, NY.
- Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó. Séaghdha, Sebastian Padó, et al. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval'10)*. 33–38.
- Geoffrey E. Hinton. 2009. Deep belief networks. Revision #91189. *Scholarpedia* 4, 5 (2009), 5947.
- Geoffrey E. Hinton, James L. McClelland, and David E. Rumelhart. 1986. Distributed representations. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol. 1. MIT Press, Cambridge, MA, 77–109.
- Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 7 (2006), 1527–1554.
- Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006), 504–507.
- Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*. 50–57.
- Liangjie Hong and Brian D. Davison. 2010. Empirical study of topic modeling in Twitter. In *Proceedings of the SIGKDD Workshop on Social Media Analytics*. 80–88.
- Amy K. Hoover, Paul A. Szerlip, Marie E. Norton, Trevor A. Brindle, Zachary Merritt, and Kenneth O. Stanley. 2012. Generating a complete multipart musical composition from a single monophonic melody with functional scaffolding. In *Proceedings of the 3rd International Conference on Computational Creativity*. 111–118.
- Dominik Hornel and Tomas Ragg. 1996. Learning musical structure and style by recognition, prediction and evolution. In *Proceedings of the 1996 International Computer Music Conference (ICMC'96)*. 59–62.
- Andrew Horner and David Goldberg. 1991. Genetic algorithms and computer-assisted music composition. In *Proceedings of the 4th International Conference on Genetic Algorithms (ICGA'91)*. 427–441.
- Ping Hou, Broes de Cat, and Marc Denecker. 2010. Fo(Fd): Extending classical logic with rule-based fixpoint definitions. *Theory and Practice of Logic Programming* 10, 4–6 (2010), 581–596.
- Taehyun Hwang and Rui Kuang. 2010. A heterogeneous label propagation algorithm for disease gene discovery. In *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM'10)*. 583–594.
- Ashwin Ittoo and Gosse Bouma. 2011. Extracting explicit and implicit causal relations from sparse, domain-specific texts. In *Proceedings of the 16th International Conference on Natural Language Processing and Information Systems (NLDB'11)*. 52–63.
- Ashwin Ittoo, Gosse Bouma, Laura Maruster, and Hans Wortmann. 2010. Extracting meronymy relations from domain-specific, textual corporate databases. In *Proceedings of the 15th International Conference on Applications of Natural Language to Information Systems (NLDB'10)*. 48–59.
- Gerhard Jäger. 2010. Natural color categories are convex sets. In *Proceedings of Logic, Language and Meaning: 17th Amsterdam Colloquium*. 11–20.

- Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. 1999. Data clustering: A review. *ACM Computing Surveys* 31, 3 (1999), 264–323.
- Grega Jakus, Veljko Milutinovic, Sanida Omerović, and Sašo Tomažič. 2013. *Concepts, Ontologies, and Knowledge Representation*. Springer-Verlag New York, NY.
- Glen Jeh and Jennifer Widom. 2002. SimRank: A measure of structural-context similarity. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*. 538–543.
- Kyle E. Jennings. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20, 4 (Nov. 2010), 489–501.
- Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. 2010. Graph regularized transductive classification on heterogeneous information networks. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part I (ECML PKDD'10)*. 570–586.
- Bradley E. Johanson and Riccardo Poli. 1998. GP-Music: An interactive genetic programming system for music generation with automated fitness raters. In *Proceedings of the 3rd Annual Conference on Genetic Programming (GP'98)*. 181–186.
- Matjaž Jursič, Borut Sluban, Bojan Cestnik, Miha Grčar, and Nada Lavrač. 2012. Bridging concept identification for constructing information networks from text documents. In *Bisociative Knowledge Discovery: An Introduction to Concept, Algorithms, Tools, and Applications*, M. R. Berthold (Ed.). Springer, 66–90.
- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*. 119–126.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 655–665.
- George Karabatis, Zhiyuan Chen, Vandana P. Janeja, Tania Lobo, Monish Advani, Mikael Lindvall, et al. 2009. Using semantic networks and context in search for relevant software engineering artifacts. In *Journal on Data Semantics XIV, S. Spaccapietra and L. Delcambre (Eds.)*. Lecture Notes in Computer Science, Vol. 5880. Springer, 74–104.
- Christopher S. G. Khoo, Syin Chan, and Yun Niu. 2000. Extracting causal knowledge from a medical database using graphical patterns. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL'00)*. 336–343.
- Douwe Kiela and Stephen Clark. 2014. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC'14)*.
- Arthur Koestler. 1964. *The Act of Creation*. MacMillan, New York, NY.
- Risi I. Kondor and John D. Lafferty. 2002. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the 19th International Conference on Machine Learning (ICML'02)*. 315–322.
- Sotiris B. Kotsiantis. 2007. Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering*. 3–24.
- Zornitsa Kozareva and Eduard Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the Web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP'10)*. 1110–1118.
- Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of the 2008 Annual Meeting of the Association for Computational Linguistics: Human Languages Technologies (ACL-08: HLT)*. 1048–1056.
- Ben Kröse and Patrick van der Smagt. 1993. *An Introduction to Neural Networks* (5th ed.). University of Amsterdam.
- Oliver Kutz, Till Mossakowski, Joana Hois, Mehul Bhatt, and John Bateman. 2012. Ontological blending in DOL. In *Proceedings of the 1st International ECAI Workshop on Computational Creativity, Concept Invention, and General Intelligence (C3GI'12)*.
- Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes, Special Issue on Quantitative Approaches to Semantic Knowledge Representations* 25, 2–3 (1998), 259–284.
- Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.
- Douglas Lenat. 1995. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38 (1995), 33–38.
- Robert P. Levy. 2001. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the 4th International Conference on Case-Based Reasoning (ICCB'01)*.
- Shu-Hsien Liao. 2003. Knowledge management technologies and applications—literature review from 1995 to 2002. *Expert Systems With Applications* 25, 2 (2003), 155–164.
- Antonios Liapis, Héctor P. Martínez, and Georgios N. Julian Togelius. 2013. Transforming exploratory creativity with DeLeNoX. In *Proceedings of the 4th International Conference on Computational Creativity (ICCC'13)*. 56–63.
- Dekang Lin and Patrick Pantel. 2001. DIRT @SBT@Discovery of inference rules from text. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*. 323–328.
- Hugo Liu and Push Singh. 2004. ConceptNet—A practical commonsense reasoning tool-kit. *BT Technology* 22, 4 (2004), 211–226.

- Maria Teresa Llano, Simon Colton, Rose Hepworth, and Jeremy Gow. 2016. Automated fictional ideation via knowledge base manipulation. *Cognitive Computation* 8, 2 (2016), 153–174.
- Ziyu Lu, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2016. Sentiment-based topic suggestion for micro-reviews. In *Proceedings of the 10th International AAAI Conference on Web and Social Media (ICWSM'16)*. 231–240.
- Penousal Machado and Amílcar Cardoso. 2002. All the truth about NEvAr. *Applied Intelligence: Special Issue on Creative Systems* 16, 2 (2002), 101–119.
- Penousal Machado, Henrique Nunes, and Juan Romero. 2010. Graph-based evolution of visual languages. In *Applications of Evolutionary Computation*, C. Di Chio, A. Brabazon, G. A. Di Caro, M. Ebner, M. Farooq, A. Fink, et al. (Eds.). Lecture Notes in Computer Science, Vol. 6025. Springer Berlin Heidelberg, 271–280.
- Penousal Machado, Juan Romero, and Bill Manaris. 2008. Experiments in computational aesthetics. In *The Art of Artificial Evolution*, J. Romero and P. Machado (Eds.). Springer, Berlin, Germany, 381–415.
- Penousal Machado, Juan Romero, Antonino Santos, Amílcar Cardoso, and Alejandro Pazos. 2007. On the development of evolutionary artificial artists. *Computers and Graphics* 31, 6 (2007), 818–826.
- Mary L. Maher, Douglas Fisher, and Kate Brady. 2013. Computational models of surprise as a mechanism for evaluating creative design. In *Proceedings of the 4th International Conference on Computational Creativity (ICCC'13)*. 147–151.
- Pedro Martins, Tanja Urbančič, Senja Pollak, Nada Lavrac, and Amílcar Cardoso. 2015. The good, the bad, and the AHA! blends. In *Proceedings of the 6th International Conference on Computational Creativity (ICCC'15)*.
- Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL'12)*.
- Jon McCormack. 1993. Interactive evolution of l-system grammars for computer graphics modelling. In *Complex Systems: From Biology to Computation*. ISO Press, 118–130.
- Jon McCormack. 1996. Grammar based music composition. *Complex Systems* 96 (1996), 321–336.
- James McDermott. 2013. Graph grammars for evolutionary 3D design. *Genetic Programming and Evolvable Machines* 14, 3 (2013), 369–393.
- Stephen McGregor, Kat Agres, Matthew Purver, and Geraint A. Wiggins. 2015. From distributional semantics to conceptual spaces: A novel computational method for concept creation. *Journal of Artificial General Intelligence* 6 (2015), 55–86.
- Stephen McGregor, Geraint A. Wiggins, and Matthew Purver. 2014. Computational creativity: A philosophical approach, and an approach to philosophy. In *Proceedings of the 5th International Conference on Computational Creativity (ICCC'14)*.
- Sarnoff A. Mednick. 1962. The associative basis of the creative process. *Psychological Review* 69, 3 (1962), 220–232.
- Larry R. Medsker and Lakhmi C. Jain. 1999. *Recurrent Neural Networks: Design and Applications*. CRC Press, Boca Raton, FL.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the Workshop at the International Conference on Learning Representations 2013 (ICLR'13)*.
- Dragana Miljkovic, Tjaša Stare, Igor Mozetič, Vid Podpečan, Marko Petek, Kamil Witek, et al. 2012. Signalling network construction for modelling plant defence response. *PLoS One* 7 (2012), e51822–1e51822–18.
- George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography* 3, 4 (1990), 235–244.
- Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP—Volume 2 (ACL'09)*. 1003–1011.
- Nicolas Monmarché, Isabelle Mahnich, and Mohamed Slimane. 2007. Artificial art made by artificial ants. In *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, J. Romero and P. Machado (Eds.). Springer, Berlin, Germany, 227–247.
- Kristine Monteith, Tony Martinez, and Dan Ventura. 2010. Automatic generation of music for inducing emotive response. In *Proceedings of the 1st International Conference on Computational Creativity*. 140–149.
- Richard Morris, Scott Burton, Paul Bodily, and Dan Ventura. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the 3rd International Conference on Computational Creativity (ICCC'12)*. 119–125.
- Vivi Nastase, Preslav Nakov, Diarmuid ó Séaghdha, and Stan Szpakowicz. 2013. *Semantic Relations Between Nominals*. Morgan & Claypool.
- Roberto Navigli. 2016. Ontologies. In *The Oxford Handbook of Computational Linguistics* (2nd ed.), R. Mitkov (Ed.). Oxford University Press.
- Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 1318–1327.
- Thien Hai Nguyen and Kiyooki Shirai. 2015. Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1354–1364.

- David Norton, Derrall Heath, and Dan Ventura. 2010. Establishing appreciation in a creative system. In *Proceedings of the 1st International Conference on Computational Creativity (ICCC'10)*. 26–35.
- Andrew Olney and Zhiqiang Cai. 2005. An orthonormal basis for topic segmentation in tutorial dialogue. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP'05)*. 971–978.
- Alex F. Osborn. 1953. *Applied Imagination*. Scribner.
- Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (ACL'06)*. 113–120.
- Patrick Pantel and Deepak Ravichandran. 2004. Automatically labeling semantic classes. In *Proceedings of the 2004 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'04)*. 321–328.
- Michael J. Paul and Mark Dredze. 2014. Discovering health topics in social media using topic models. *PLoS One* 9, 8 (2014), e103408.
- Adam Pease and Ian Niles. 2002. IEEE standard upper ontology: A progress report. *Knowledge Engineering Review* 17, 1 (2002), 65–70.
- Michel Plantié and Michel Crampes. 2013. Survey on social community detection. In *Social Media Retrieval*, N. Ramzan, R. van Zwol, J.-S. Lee, K. Clüver, and X.-S. Hua (Eds.). Springer, London, England, 65–85.
- Jeffrey Putnam. 1994. *Genetic Programming of Music*. Technical Report. New Mexico Institute of Mining and Technology.
- V. V. Raghavan and S. K. M. Wong. 1986. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science* 37, 5 (1986), 279–287.
- Martin Raubal. 2004. Formalizing conceptual spaces. In *Proceedings of the 3rd International Conference on Formal Ontology in Information Systems (FOIS'04)*. 153–164.
- Martin Raubal. 2008. Cognitive modeling with conceptual spaces. In *Proceedings of the Workshop on Cognitive Models of Human Spatial Reasoning*. 7–11.
- Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*. 41–47.
- John T. Rickard. 2006. A concept geometry for conceptual spaces. *Fuzzy Optimization and Decision Making* 5, 4 (2006), 311–329.
- John T. Rickard, Janet Aisbett, and Greg Gibbon. 2007a. Reformulation of the theory of conceptual spaces. *Information Sciences* 177, 21 (2007), 4539–4565.
- John T. Rickard, Janet Aisbett, and Greg Gibbon. 2007b. Knowledge representation and reasoning in conceptual spaces. In *Proceedings of the 1st IEEE Symposium on Foundations of Computational Intelligence (FOCI'07)*. 583–590.
- Steven Rooke. 1996. The Evolutionary Art of Steven Rooke. Retrieved June 1, 2014 from <http://srooke.com>.
- Nicholas Roussopoulos and John Mylopoulos. 1975. Using semantic networks for data base management. In *Proceedings of the 1st International Conference on Very Large Data Bases (VLDB'75)*. 144–172.
- Gerard Salton. 1971. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Upper Saddle River, NJ.
- Gerard Salton. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison Wesley Longman, Boston, MA.
- Sunita Sarawagi. 2008. Information extraction. *Foundations and Trends in Databases* 1, 3 (2008), 261–377.
- Rob Saunders and John Gero. 2001. The digital clockwork muse: A computational model of aesthetic evolution. In *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in Arts and Science (AISB'01)*. 12–21.
- Angela Schwering and Martin Raubal. 2005. Spatial relations for semantic similarity measurement. In *Perspectives in Conceptual Modeling*, J. Akoka, S. W. Liddle, I.-Y. Song, M. Bertolotto, I. Comyn-Wattiau, W.-J. Heuvel, et al. (Eds.). Lecture Notes in Computer Science, Vol. 3770. Springer, 259–269.
- Thomas Serre, Gabriel Kreiman, Minjoon Kouh, Charles Cadieu, Ulf Knoblich, and Tomaso Poggio. 2007. A quantitative theory of immediate visual recognition. *Progress in Brain Research* 165, (2007), 33–56.
- Karl Sims. 1991. Artificial evolution for computer graphics. *ACM Computer Graphics* 25 (1991), 319–328.
- Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wanli Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, R. Meersman and Z. Tari (Eds.). Lecture Notes in Computer Science, Vol. 2519. Springer, 1223–1237.
- Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman, and William T. Freeman. 2005. Discovering objects and their location in images. In *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV'05)*, Vol. 1. 370–377.
- Benjamin D. Smith and Guy E. Garnett. 2012. Improvising musical structure with hierarchical neural nets. In *Proceedings of the 8th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE'12)*. 63–67.

- Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA, 1297–1304.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, et al. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*. 1631–1642.
- John F. Sowa. 1992. Semantic networks. In *Encyclopedia of Artificial Intelligence* (2nd ed.), S. C. Shapiro (Ed.). John Wiley & Sons, New York, NY.
- John F. Sowa. 2000. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA.
- John F. Sowa. 2010. Ontology. Retrieved June 1, 2014 from <http://www.jfsowa.com/ontology>.
- Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28, 1 (1972), 11–21.
- Lee Spector and Adam Alpern. 1994. Criticism, culture, and the automatic generation of artworks. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI'94)*. 3–8.
- Carlo Strapparava, Alessandro Valitutti, and Oliviero Stock. 2007. Automatizing two creative functions for advertising. In *Proceedings of the 4th International Joint Workshop on Computational Creativity*. 99–108.
- Ron Sun. 2008. *The Cambridge Handbook of Computational Psychology*. Cambridge University Press.
- Yizhou Sun and Jiawei Han. 2012. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool.
- Yizhou Sun and Jiawei Han. 2013. Mining heterogeneous information networks: A structural analysis approach. *ACM SIGKDD Exploration Newsletter* 14, 2 (2013), 20–28.
- Yizhou Sun, Yintao Yu, and Jiawei Han. 2009. Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. 797–806.
- Don R. Swanson. 1990. Medical literature as a potential source of new knowledge. *Bulletin of the Medical Library Association* 78, 1 (1990), 29–37.
- Asuka Terai and Masanori Nakagawa. 2009. A neural network model of metaphor generation with dynamic interaction. In *Proceedings of the 23rd International Conference on Artificial Neural Networks (ICANN'09)*. 779–788.
- Miles Thorogood and Philippe Pasquier. 2013. Computationally created soundscapes with audio metaphor. In *Proceedings of the 4th International Conference on Computational Creativity (ICCC'13)*. 1–7.
- Peter M. Todd. 1992. A connectionist system for exploring melody space. In *Proceedings of the International Computer Music Conference (ICMC'92)*. 65–68.
- Hannu Toivonen and Oskar Gross. 2015. Data mining and machine learning in computational creativity. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5, 6 (2015), 265–275.
- Nao Tokui and Hitoshi Iba. 2000. Music composition with interactive evolutionary computation. In *Proceedings of the 3rd International Conference on Generative Art (GA'00)*, Vol. 17. 215–226.
- Peter D. Turney. 2007. *Empirical Evaluation of Four Tensor Decomposition Algorithms*. Technical Report ERB-1152. National Research Council, Institute for Information Technology.
- Peter D. Turney and Michael Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems* 21, 4 (2003), 315–346.
- Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Artificial Intelligence Research* 37, 1 (2010), 141–188.
- Tatsuo Unemi. 1999. SBART2.4: Breeding 2D CG images and movies, and creating a type of collage. In *Proceedings of the 3rd International Conference on Knowledge-Based Intelligent Information Engineering Systems (KES'99)*. 288–291.
- Frank van der Velde. 1993. Is the brain an effective turing machine or a finite-state machine? *Psychological Research* 55 (1993), 71–79.
- Frank van der Velde. 2015. Communication, concepts and grounding. *Neural Networks* 62 (2015), 112–117.
- Frank van der Velde and Marc de Kamps. 2006. Neural blackboard architectures of combinatorial structures in cognition. *Behavioral and Brain Sciences* 29 (2006), 37–70.
- Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter (Eds.). 2008. *Handbook of Knowledge Representation*. Elsevier, Amsterdam, Netherlands.
- Tony Veale. 2012. From conceptual “mash-ups” to “bad-ass” blends: A robust computational model of conceptual blending. In *Proceedings of the 3rd International Conference on Computational Creativity (ICCC'12)*. 1–8.
- Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics* 39, 3 (2013), 665–707.
- Chris Venour, Graeme Ritchie, and Chris Mellish. 2010. Quantifying humorous lexical incongruity. In *Proceedings of the 1st International Conference on Computational Creativity (ICCC'10)*. 268–277.

- Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, et al. 2016. Multi-domain neural network language generation for spoken dialogue systems. In *Proceedings of the 2016 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT'16)*.
- Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. 2010. Twiterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM'10)*. 261–270.
- Geraint A. Wiggins. 2006a. A preliminary framework for description, analysis and comparison of creative systems. *Journal of Knowledge Based Systems* 19, 7 (2006), 449–458.
- Geraint A. Wiggins. 2006b. Searching for computational creativity. *New Generation Computing* 24, 3 (2006), 209–222.
- M. Tsan Wong and A. Honwai Chun. 2008. Automatic haiku generation using VSM. In *Proceedings of the 7th WSEAS International Conference on Applied Computer and Applied Computational Science (ACACOS'08)*. 318–323.
- S. K. M. Wong, W. Ziarko, and P. C. N. Wong. 1985. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'85)*. 18–25.
- Ping Xiao and Josep Blat. 2013. Generating apt metaphor ideas for pictorial advertisements. In *Proceedings of the 4th International Conference on Computational Creativity (ICCC'13)*. 8–15.
- Bo Yang, Dayou Liu, and Jiming Liu. 2010. Discovering communities from social networks: Methodologies and applications. In *Handbook of Social Network Technologies and Applications*, B. Furht (Ed.). Springer USA, 331–346.
- Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, Vol. 16. 321–328.
- Song-Chun Zhu and David Mumford. 2007. A stochastic grammar of images. *Foundations and Trends® in Computer Graphics and Vision* 2, 4 (2007), 259–362.
- Hai Zhuge. 2012. The semantic link network. In *The Knowledge Grid: Toward Cyber-Physical Society* (2nd ed.). World Scientific Publishing Co., 65–210.

Received September 2015; revised November 2017; accepted February 2018