Chapter 1
Dimensions of semantic similarity

Paweł Szmeja, Maria Ganzha, Marcin Paprzycki, Wiesław Pawłowski

Abstract Semantic similarity is a broad term used to describe many tools, models and methods applied in knowledge bases, semantic graphs, text disambiguation, ontology matching and more. Because of such broad scope it is, in a "general" case, difficult to properly capture and formalize. So far, many models and algorithms have been proposed that, albeit often very different in design and implementation, produce a single score (a number) each. These scores come under the single term of *semantic similarity*. Whether one is comparing documents, ontologies, entities, or terms, existing methods often propose a *universal* score—a single number that "captures all aspects of similarity". In opposition to this approach, we claim that there are many ways, in which semantic entities can be similar. We propose a division of knowledge (and, consequently, similarity) into categories (*dimensions*) of semantic relationships. Each *dimension* represents a different "type" of similarity and its implementation is guided by an interpretation of the *meaning* (semantics) of that similarity score in a particular dimension. Our proposal allows to add extra information to the similarity score, and to highlight differences and similarities between results of existing methods.

Paweł Szmeja
Systems Research Institute, Polish Academy of Sciences
e-mail: Pawel.Szmeja@ibspan.waw.pl

Maria Ganzha
Systems Research Institute, Polish Academy of Sciences; Warsaw University of Technology
e-mail: Maria.Ganzha@ibspan.waw.pl

Marcin Paprzycki
Systems Research Institute, Polish Academy of Sciences; Warsaw Management Academy
e-mail: Marcin.Paprzycki@ibspan.waw.pl

Wiesław Pawłowski
Faculty of Mathematics, Physics, and Informatics, University of Gdańsk; Systems Research Institute, Polish Academy of Sciences
e-mail: Wieslaw.Pawlowski@inf.ug.edu.pl

1.1 Introduction

Semantic similarity, understood broadly, has been applied in very different fields such as psychology, linguistics, biology, knowledge modeling, artificial intelligence, and others. Even though, in our work, we focus on computer science (and mathematics), the understanding of similarity in any domain is influenced by other domains. Within the scope of computer science there are many areas of interest for similarity scoring, such as graphics (e.g. face recognition), information retrieval, machine learning, etc. In this context, *semantic* similarity algorithms are focused mostly on computational linguistics and semantic reasoning, each with multiple applications. The most popular direct areas of application are ontology matching and document (i.e. text) similarity scoring.

To introduce some order into our considerations, let us declare specific *objects* that we consider most relevant (in the scope of this work). Those are *documents* (natural language texts organized in corpora), *terms* (e.g. atomic parts of a text), *ontologies* (a representation of a knowledge base, e.g. a semantic graph) and *entities* (atomic parts of a knowledge base or an ontology). We also look at *entity descriptions* contained in ontologies. We mostly consider pairwise comparisons between any two objects of the same type (e.g. two documents, and not one document and one ontology). However, to avoid gratuitous verbosity, the focus is on comparison of entities and their descriptions. Nevertheless, this paper presents a theoretical approach, or a "meta-model" of semantic similarity and the presented ideas can be applied in different fields where similarity is relevant and to objects other than entities.

To formulate specific examples we use OWL [4] (the most popular ontology description language) and, occasionally, *description logic* formulas. This is done in order to illustrate practical applicability of our ideas.

This paper is an extension and continuation of our previous work [1], where we have briefly introduced and justified the idea of semantic similarity dimensions.

We proceed as follows. Section 1.2 contains a short introduction to relevant concepts from the description logic. In following sections 1.3 and 1.4 we summarize the existing approaches (both theoretical and practical) to calculating semantic similarity. We also briefly explore the general truths about similarity. Semantic *similarity dimensions* are introduced in section 1.5. Its subsections present archetypes of *dimensions*, along with general information and examples. Section 1.6 illustrates practical application of the *dimensional* similarity method in the field of ontologies, while section 1.7 confronts the *dimensional* similarity score and results of other similarity methods. Properties of *similarity dimensions* are examined in section 1.8, while section 1.9 outlines more general use cases. Finally section 1.10 presents a summarized case for *semantic similarity dimensions*.

1.2 Description logic

Let us start with a brief introduction to description logic (DL). Information presented here is needed in order to understand some examples given in later sections. This is because OWL is based on DL, and OWL axioms can be written in the form of mathematical formulas expressed in DL. This representation gives a useful perspective on ontological entities.

In description logic, knowledge is stored in *knowledge bases* that contain *axioms* (also called *facts*). An ontology is a collection that contains knowledge and, within the scope of this paper, is considered to be an equivalent of a knowledge base. More formally, a knowledge base is part of a mathematical model and ontology document (specifically in computer science) is a presentation of this model.

Each knowledge base (KB) (and each ontology) can be partitioned into *TBox*, *ABox* and *RBox*. Each of those boxes contains different kinds of axioms. This division extends to different *entities* and *entity descriptions*.

The TBox contains *concepts* (also called *classes*), i.e. the declarations and descriptions of concepts. A *declaration* is simply a statement about the kind of an *entity* (in this case—a class), and each *description* of a concept is constructed from *concept names*, *role names*, constants and a set of DL *constructors*. The *description* is said to provide the explanation of a semantic meaning of a class.

A set of classes, organized into a hierarchy, is called a *taxonomy*. In taxonomy classes may have descendants (specializations) and ancestors (*subsumers* or generalizations). A relation of subsumption in many semantic graphs has the name *IS-A* e.g. “computer IS-A machine”. The IS-A relations occur very often and form directed sub-graphs in semantic graphs. In DL every taxonomy contains a special class—*Thing* ($\top$), that is the “top” of the hierarchy, i.e. it has no ancestors. Moreover each concept is necessarily a descendant of $\top$, which means that every concept in an ontology is a part of the same hierarchy. In other words every concept is of type *Thing* and, in semantic graphs, there exists at least one path along the IS-A relation between any two concepts. *Thing* is considered to be the root of a taxonomy tree. In this paper, concepts with rich *descriptions* are called *complex*, as opposed to simple concepts, descriptions of which consist only of class names and describe only the concepts’ position within the taxonomy. Historically, simple concept descriptions are important, because many old ontologies were formulated exclusively in terms of taxonomies, and, consequently, some similarity algorithms consider only the taxonomic part of ontologies. Taxonomies and classes are still considered central to a lot of ontologies.

The ABox contains *declarations* and *descriptions* of *individuals*. *Individuals* are *instances* of classes, i.e. each instance is of at least one *type* and, necessarily, of type *Thing*. A description of an *individual* is comprised of *types* and *assertions* about *properties* built from *concept* names, *role* names and constants. Property assertions are parts of a description that are specific to each individual and together with individual’s types describe the meaning of the entity. Individuals usually do not form a hierarchy. They are, nevertheless, strongly tied (via the *types*) to the taxonomy. As a

consequence of the meaning of subsumption, an individual that is explicitly of type A is also of all types that are ancestors of A (including $\top$).

The RBox contains *declarations* and *descriptions* of *roles* (also called *properties*). A role description defines the role's domain, range and characteristics (such as symmetry, transitivity and others), which likens roles to mathematical binary relations. In OWL DL [4], an OWL variant,two types of roles are distinguished: *object* properties and *data* properties. The range of an object property is a class, while the range of a data property is a literal (e.g. a numerical value or a string). For both types of roles the domain is a concept. The RBox is often considered to be a part of ABox, as opposed to a "box" of its own. In any case the main line of division of a knowledge base lies between ABox and TBox.

Following the division of KB into "boxes", within the scope of this paper the term (ontological) "*entities*" refers to *concepts*, *individuals* or *properties*. The most relevant similarity calculation is between entities from the same "box", although comparison between, for instance, a class and an individual is also, theoretically, possible (we touch upon this later).

Different varieties of DL (sometimes called *profiles*) determine what *constructors* are available when formulating axioms, as well as what syntactic variant is allowed (i.e. what symbol sequences are allowed). In order to clearly present our ideas, in this paper, most examples of DL expressions are given in a simple DL formalism $\mathscr{EL}$ (see table 1.1), unless otherwise noted. We also discuss how the ideas may be extended to more expressive description logics.

Table 1.1 DL constructors

$\mathscr{EL}$	Name	Syntax
$*$	top concept	$\top$
	bottom concept	$\bot$
	concept (class)	C
	concept negation	$\neg C$
$*$	concept conjunction	$C_1 \sqcap C_2$
	concept disjunction	$C_1 \sqcup C_2$
$*$	existential restriction	$\exists R.C$
	universal restriction	$\forall R.C$

In what follows, *concept* names are denoted by capital letters $A, B, C, ..$, *individual* names by lowercase letters $a, b, .., o$, and *role* names by lowercase letters $p, r, .., z$. Each name may have an optional index, e.g. C_1. An expression $C(a)$ means that individual a is of type C. Expression $r(a,b)$ is a role assertion and denotes that a is related to b by role r, a being the realization of the roles domain and b its range. Table 1.1 summarizes relevant DL constructors. Constructors in $\mathscr{EL}$ are (by definition) limited to: top concept, concept conjunction and existential quantification (restriction). For more details about DL, its constructors, semantics, and varieties refer to [40].

Some relations defined in DL are of special importance to similarity scoring. Those are primarily subsumption ($\sqsupseteq$), inclusion ($\sqsubseteq$) and equivalence ($\equiv$). The $\sqsubseteq$

corresponds directly to the IS-A relation, and the $\sqsupseteq$ combined with the $\sqsubseteq$ means the same as the $\equiv$. While subsumption is usually reserved for classes, the equivalence relation can be applied to any entity and has special interpretation of "maximal similarity". Unsurprisingly, when two entities are equivalent, they should be treated as one and the same entity, and their similarity score should be maximal. In similar fashion, a negated ($\neg$) entity should be maximally different from the entity it negates, but most often negation is neither part of the DL profile of an ontology, nor is considered when calculating similarity.

There are two definitions that are highly relevant to calculation of similarity, that can be expressed in DL—the *least common subsumer* [41] and the *most specific concept* [42].

The *least common subsumer* (LCS) of two entities X and Y is the most specific (i.e. farthest from the root) entity that is an ancestor to both X and Y. In a taxonomy, it is a concept that shares the most types with compared concepts, and itself is a type (generalization) of compared concepts. In other words, the LCS is a class that is a superclass to both X and Y. Since there may be multiple such superclasses, we choose the one that is least general, i.e. as far from the *Thing* and as close to X and Y as possible. Since LCS represents, in a sense, the most information the entities have in common, it is sometimes treated as the central part of similarity algorithms.

The *most specific concept* (MSC) of an individual is a concept whose description is built from assertions about the individual in a way that includes every such assertion. In other words it is a class that is built specifically to contain an individual and in its construction is guided by the description of the individual. It is often so specific that it only contains the one individual it was constructed for. The process of construction of the MSC utilizes standard semantic deduction [57] and is described in detail in [56]. The MSC is, in general case, not unique and, because of that, its usefulness is put into question. The details are beyond the scope of this work and we refer interested readers to [56]. Even more information about both LCS and MSC can be found in [40, 41, 42]. In further sections we give specific examples for LCSs.

1.3 Similarity

There is a multitude of works dedicated to similarity across multiple domains, including psychology and sociology [86, 87], as well as more technical fields, such as mathematics, computer science, or engineering (i.e. similitude [88]). To keep the text coherent, and not to stray too far from the core ideas, in this section we present some general observations about similarity, relevant to the main content of this work. Since computer science is the main focus, articles relevant to this field of science are referenced throughout the text. Let us now start our general considerations.

Features of similarity

There are many properties that may apply to a similarity measure. Most measures reflect the following two general observations about similarity:

- It grows with the commonality of objects
- It decreases with the difference between objects

Those two terse and laconic statements are ones of very few that, for all practical purposes, can be applied to vast majority of formalized similarity measures. Whether we compare documents, terms, ontologies or other objects, the factors that increase similarity can be summarized as the *commonality* between objects. What contributes to decrease of similarity is the *disparity*, often included in the score implicitly, as opposite to the *commonality*.

Here, it should be stressed that, although other observations can be made about similarity, there is no strong/community-wide consensus, based on which, one could construct a general definition. Furthermore, there are a number of "features" of similarity, both formal and informal, that have support and opposition. A notable informal feature of similarity is that, in human judgment, common features carry more weight than disparate ones [48]. This is exemplified in measures that explicitly consider only *commonality* and discard differences, i.e. work under the supposition that the initial similarity is minimal (zero) and each common feature increases it. A different approach relies on a "balancing" between commonalities and disparities, where each of those increase and decrease similarity (respectively) that, initially, is set at a middle score (e.g. 0.5 on a scale from 0 to 1).

Attentiveness

Another concept relevant to similarity is classification that stems from computer graphics, and divides similarities into "Pre-attentive" and "attentive" [89, 90]. Pre-attentive similarity is measured before "interpretation" of entities (or "stimuli", in graphics processing terminology), while attentive methods are used after entities have been interpreted, classified and put into context. The "interpretation" process, while specific to computer graphics, can be extended to semantic similarity in general. At first glance, semantic similarity falls squarely into the attentive category, because semantic descriptions or features are already an interpretation of entities. On the other hand, similarity algorithms commonly do not distinguish between possible different interpretations of the same entity. This is highly relevant in graphic databases, where, for instance, searching for images most similar (to a reference image) requires different queries, depending on whether we are interested in shape or color palette similarity. Other, advanced features of an image may be calculated, such as painting style, and they all require attentive methods. Extending this idea into general semantic similarity, we may add similarity scoring with respect to provenance (e.g. authorship) information for the same images. The different "views"

on similarity are, from the point of view of attentive methods, different interpretations of the same entities. This idea is expanded upon, although with different terminology, in later sections, where we describe semantic similarity dimensions.

Reference similarities

Another notable point is that similarity often depends on the *context*, where each feature has a weight based on *subjective* evaluation of importance (i.e. opinion) and circumstances surrounding the comparison. In human judgment, knowledge of the person performing the comparison is very important [53] and is de facto the implicit "knowledge base" that we score against. However, it is not directly relevant in case of comparison of DL entities, because, the knowledge used for such comparison is explicitly defined and available in KB. It has, however, heavy bearing on the quality of similarity methods, because their results are often judged against human (expert) opinions. In this way, the evaluation of a similarity algorithm is dependent not only on the expert doing the evaluation, but also on the quality of the ontology.

There are reference sets of similarities, such as the Miller's benchmark [43]. The benchmark itself (building on previous work [53]) produced a set of 30 pairs of generic terms along with a similarity score averaged from judgments of 38 students. It was conceived as a way to gain insight into how humans score similarity. The terms were chosen to be purposefully ambiguous, which adds another layer of (disambiguation) challenge for the similarity algorithms. Moreover, such benchmarks are not a good reference point for big ontologies, since those are usually detailed and contain expert knowledge of a certain domain, which means that there are no ambiguous terms in them. Later, the benchmark was reproduced [50] with considerably different results, which further questions its validity as a similarity algorithm evaluation tool. Nevertheless, many works refer to this specific benchmark as a reference and proof of good (i.e. correct) results [44, 45, 46, 47, 91].

There are organized efforts to counteract the difficulties in evaluating (semi-)automated methods of similarity scoring, such as OAEI (Ontology Alignment Evaluation Initiative) [8]. This initiative is dedicated specifically to evaluating only ontology alignment tools and its evaluations are organized into yearly editions, each with many "tracks". In each track there are some reference ontologies and alignments prepared. By ensuring that both ontologies and reference alignments are of good quality, OAEI is able to, at least in principle, provide more meaningful evaluation results, than simple comparison to a benchmark. Another approach was adopted by Reuters, which publishes corpora of documents for text categorization, e.g. Reuters-21578 or RCV1 (Reuters Corpus Volume 1). Instead of declaring an authoritative reference categorization, the corpora may be used to compare results of different methods, or to test improvement of a single method over its previous iteration. Other similar corpora exist, e.g. Ohsumed [92] for medical documents and 20NewsGroups [93] for newsgroup documents.

Granularity

It is worth noting that granularity of information has an influence on similarity. Informally, the more details we include in our comparison, the less impact on similarity each of them will have. Consequently, a general feature of observation about an object has a bigger weight than a detailed description of the same feature. In other words, an expert (or an expert ontology) has a different (more detailed) view than a layman (a general ontology). In case of human judgment, but also algorithms modeled after it, taking into account that *commonality* carries more weight, the similarity of a feature is likely to decrease, when we take a closer look at it. For example, a simple property, such as age of people, might be exactly the same, when we look only at the birth year, but will decrease with the increase of accuracy to days or even minutes. For a layman, creatures such as monkeys and chimpanzees would be much more similar, than in the eyes of an expert. A more general observation is that the amount of information we have about an entity greatly influences similarity, which is most pronounced in probabilistic methods, such as in [50], where similarity can change, even if we add knowledge seemingly unrelated to the compared objects. In particular, granularity of information may strongly influence human performed quality evaluation of similarity measures (cf. [43]).

Distance and closeness

Similarity is often considered in the context of distance between entities. The idea comes from psychology [86] and states that entities may be put into a multidimensional space, where each dimension is a separate characteristic. In such theoretical space, the distance between entities is the evidence of dissimilarity, which is, informally, an inverse of similarity. While in specific applications it may be possible to construct a finite set of dimensions [90], in general, the sheer number of possible characteristics in semantic descriptions makes construction of general algorithms based on this idea difficult [25]. The distance-based similarity is also applicable to graph structures (more on this in later sections).

Similarity ordering

It has been argued that just an ordering of concepts, with respect to similarity, is more useful than a number. In many applications one is primarily interested in finding an entity that is closest to a reference entity, the actual value of "closeness" being secondary. In case of three objects, one reference object and two compared objects, we might be satisfied just knowing, which of the comparison objects is more similar to the reference object, rather than learning the numerical score. This is clearly pronounced in the difference between regular search and search based on similarity

measures. The first is a partition of the search space into entities that fit the search criteria (i.e. the query), and those that do not. Similarity search, essentially, responds with the complete search space, ordered by similarity. Note that, in practice, we are usually interested only in those entities that have an extreme value of similarity—either very high or very low. In both cases, the actual numerical scores, often are irrelevant, as long as they are above (or below) some threshold.

1.4 Similarity calculation methods

Let us now present a, non-exhaustive, list of selected algorithms and methods of calculation of similarity, that are both used in practice and relevant to presentation of dimensions of semantic similarity. In order to focus on presentation of our own ideas, rather than summarizing all existing similarity methods, we have chosen to describe only a few methods. For a richer list of similarity methods, see [34]. Note that the semantic measures library (SML) [85] contains implementations of a large number of methods described here, as well as many others.

Edge methods

Edge-based models work on assumption that edge distance in a graph is meaningful for similarity. Needless to say, edge-based methods require a graph structure, such as a taxonomy. Those methods view ontologies as directed graphs, where small distance along some edge type is the evidence of similarity, and long paths indicate dissimilarity. A common criticism of edge methods, when applied to ontologies, is that they work under the assumption, that each edge in a path has the same semantic distance. In practice, however, there is no formal evidence to back up this assumption, and some evidence that indicates the opposite [30, 33].

The simplest approach considers similarity to be equal to the length of the shortest path between a pair of entities (e.g. concepts) $S_{Rada}(X,Y) = \min(paths(X,Y))$ ([35]), where $paths(X,Y)$ is the set of path lengths in an IS-A graph (see, Section 1.5). More sophisticated methods, such as e.g. $Sim_{Wu}(X,Y) = \frac{2*depth(LCS(X,Y))}{depth(X)+depth(Y)}$ ([37]), involve normalization and take into account depth of the compared entities, depth of their LCS, or length of path between the root, LCS and entities. Finally, [39] utilizes multiple relations (not just IS-A) in a graph (multigraph). Because edge methods regularly use simple mathematical rations, they are often applicable to the dimensional approach to similarity, where the same formulas are used on distances along dimensions, instead of path lengths.

Feature methods

A big class of similarity algorithms are *feature* methods. In these methods, each entity is represented by a set of semantic *features* and the entity similarity is equivalent to similarity of feature sets. What the features are, and how to identify and construct them, depends on the domain of application and, sometimes, on a particular implementation. For instance, in computer graphics, a set of features for each image depends on the software, and may include shape, size, texture, color, position, etc. In general case, however, the situation is complicated and there is no universal algorithm of representing entities as a set of features. Feature methods are applied in different domains (e.g. graphics [3], reasoning [36] and others) and "features" have slightly different meaning and are constructed (or extracted) in a different way, depending on the domain model. Usually, features are crisp (not fuzzy), i.e. a feature either belongs to a feature set, or not. This approach makes it difficult to evaluate similarity of features that have numerical representations, like the aforementioned color (e.g. in the RGB color space). When put into a set, a specific numerical value of a color is, in general, an entirely different feature than any other color value, no matter how close the color are. This property is another reason for the division of feature methods into specific implementations (e.g. in graphics) and general formulas (with crisp feature sets).

In ontologies there are many ways in which *feature* sets can be constructed from a description of an entity. In case of concepts, the features are usually considered to be the concept's ancestors, its roles, instances, or a set of all of those. Details of how a complex description is converted into a set of features depend on particular method and underlying logic (see, Section 1.5 and onward). A set of *features* of an individual may be constructed from its types and role assertions. A usual approach is to use only role assertions directly mentioned in the entity definition. In such approach, the semantic descriptions with color properties (e.g. *X hasColor red* and *Y hasColor light-red*) consider the property and its value as atomic, and do not go into detail about possible similarity of *red* and *light-red*. It is worth noting that, in some specific cases, the feature sets can be constructed in a very natural way. This is the case of WordNet [61], which explicitly defines synsets (sets of synonyms) that can be, with no additional effort, treated as feature sets. Another noteworthy property is that the feature sets in a taxonomy may be defined as a IS-A neighborhood of a class. In such case, the feature method is, conceptually, very close to an edge method, because they both use very similar information as input.

In *Tversky's ratio model* [48] similarity of two sets of features X_F and Y_F is given by the formula $S_{Tv}(X_F, Y_F) = \frac{\alpha f(X_F \cap Y_F)}{\alpha f(X_F \cap Y_F) + \beta f(X_F - Y_F) + \gamma f(Y_F - X_F)}$, where $X - Y$ is a set difference (relative complement of Y in X), f is a monotonically increasing function (usually set cardinality), while α, β and γ are positive coefficients. The coefficients control importance ("weight") of common features and features exclusive to either set. For different choices of values of the coefficients, Tversky's *ratio model* has different properties and produces different formulas. In particular, for $\alpha = \beta = \gamma = 1$ and $f = |\cdot|$ the model becomes the *Jaccard index* [2] $J(X_F, Y_F) = \frac{|X_F \cap Y_F|}{|X_F \cup Y_F|}$. Some

methods ([83, 84]) include an ad-hoc weighting of coefficients for different features, instead of a fixed set of coefficients for all features. Tversky also proposed a *contrast model* represented by the formula $S_{TvC}(X_F, Y_F) = \alpha f(X_F \cap Y_F) - \beta f(X_F - Y_F) - \gamma f(Y_F - X_F)$.

Information-theoretic methods

Methods from an information-theoretic class approach similarity from the point of view of information theory [54] and assume that similarity is strictly related to the amount of information each of compared entities provides. This class is represented by *Information Content* (IC) model proposed by Resnik ([50]). IC of an entity e is computed from its *probability* $p(e)$: $IC(e) = -\log(p(e))$. When applied to a textual entity in a corpus, $p(e)$ is equal to the probability that this entity appears in a given document from the corpus. In the context of a taxonomy, probability of an entity is inversely proportional to the number of entities it subsumes. By this definition, IC is monotonically decreasing from leaves (most informative) to the root (least informative). Resnik's similarity is calculated from IC of the *most informative common ancestor* (MICA)—a common subsumer that has the maximum IC: $S_{Res}(X,Y) = IC(MICA(X,Y))$. MICA is closely related to LCS. Some works build on Resnik's approach by relaxing it's reliance on the LCS. For instance, Lin proposed a formula that involves the Information Content of the entities themselves, alongside their LCS: $S_{Lin}(X,Y) = \frac{2 \times S_{Res}(X,Y)}{IC(X)+IC(Y)}$. Other methods, such as the one proposed in [52], use only the number of immediate children as a measure of IC, where high number of children denotes low IC. Several other methods of calculating IC have been proposed [96, 95, 97], with a lot of them focusing specifically on WordNet.

Geometric methods

In the geometric approach to similarity, objects are represented as points in a multi-dimensional geometric space. Any feature of an object is converted to a number that serves as a coordinate. This approach directly corresponds to multi-dimensional similarity space described earlier. A set of coordinates represents the entire object in a space. The similarity is simply calculated as the shortest geometric (usually Euclidean) distance between two points. As such, just like any metric distance, it has the properties of minimality, and those of a mathematical metric. Unfortunately, many features are not easily subjected to conversion into a geometric dimension, as it requires them to be represented as a set of points on a continuous line. While some features have a natural representation in a geometric dimension, such as the RGB model of color, others do not, unless they are specifically designed to have that property (such as a brand of a product). A work on high-dimensional spaces [25]

describes other problems that are relevant when dealing with a high number of geometric dimensions. In practice, only domain and problem specific implementations of geometric methods exist, such as the one proposed in [90] for an image database.

Other methods

Some methods do not fit neatly into the above categories and are considered "hybrid". Such methods (e.g. [84, 51]) use characteristics from multiple categories and combine, for instance, path length with depth in taxonomy, or taxonomic neighborhood. Often, such methods use weighted sum (with weights tuned to a specific data set) of separate results for each considered perspective, or "sub-method".

A decisively syntactic (i.e. non-semantic) class of methods that deserve mention, are the edit distance methods, most prominent of which is the Hamming distance [94]. The general idea is that high number of edits that need to be done in order to transform one entity into the other is indicative of dissimilarity. Different methods define "edits" in a different way, and for strings those usually include removing a character or adding one. Edit distance is usually not applied to feature sets, although some feature set methods compute similarity score in a way reminiscent to the edit distance. It is, however, relevant to ideas presented in section 1.5.

Notice that each of the methods, presented in this section, makes some assumption as to the model of information. A feature method requires a set of features, an edge method needs a graph, etc. In order to apply each of those methods to a knowledge base we need to present it in a particular fashion—as a graph, a DL formula, a set of features and so on. In order to be applied to an ontology, each method requires a different *perspective*.

1.5 Semantic similarity dimensions

Let us now recall that, in their foundation, similarity and meaning (semantics) are inherently human concepts. From this point of view, a similarity score should have an explanation (or interpretation) that is understandable for a human. Let us consider a simple example of comparison of two physical objects. There are many *ways* in which they can be similar or dissimilar, two of them being shape and color. Those two kinds of features are independent with respect to similarity, e.g. objects can have similar color and different shape (and vice-versa). Canonical ways of automated calculation of similarity (described above) would produce a single score that would in some way combine similarity of shape and color. However, the two similarities, when treated separately, provide more information to a human, because they have a clear *interpretation*. Therefore we can assume that a person that knows this interpretation has a better understanding of how similar any two physical objects

are. In this toy example shape and color contribute to two separate *dimensions* of similarity.

From this, it can be conjectured that similarity of semantic entities has many different aspects that are being grouped together, based on what part of available data (or knowledge) is used (regardless of the actual similarity method). Those groups represent different types (dimensions) of semantic relationships and, therefore, similarity. This idea draws on the concept of *knowledge dimensions* originating from [6], where authors also divided ontological knowledge into subsets (dimensions) and applied (in [5]) to calculation of similarity in WordNet. However, there the scores from each dimension were still *combined* into a final similarity score (a single number). Our idea also borrows from geometric and feature models of similarity, and is closely related to attentiveness and multi-dimensional similarity space, described in section 1.3.

When approaching this from a different perspective, observe that approaches to either grouping attributes (features), or dividing the data into (what we call) dimensions were focused on results given by some predetermined method. In other words, *the starting point was a method* that provided foundation to interpret the result. Our approach starts from explaining the *nature / semantics* of the dimension that we are interested in, and then finds the method that would produce said result. In this way the recognized dimensions are *interpretation* driven, rather than method driven. In fact, the same method may be used in different dimensions, as exemplified in section 1.6.

Note that the concept of different kinds of similarity has been present in the literature, in one form or another, for a long time. For instance [12] contains a summary of ontology matching methods and categorizes them by the kind of data they use. Sample "kinds" of methods use comparisons of entity labels, their "attributes" (in DL terms—assertions), instances of classes, position of entities in taxonomy and others. Categorization described in [12] complements a more general work on schema matching [22] that presents its own division of matching methods by type. Later work [13] reviews methods of ontology matching and distinguishes methods that use structure of the ontologies and those that utilize entities (called *structure-level* and *element-level* dimensions). The categorization goes deeper with dimensions such as *syntactic*, *semantic*, *external*, *terminological*, *extensional* and others, some of which overlap (for more detais, see [13]). The state of the art for ontology matching, described in [103], contains more detailed descriptions of different matching methods with specific examples of implementations. Later work [14], proposes a slightly different division into language-, linguistic-, string-, and structure-based approaches. Another example is [49], which mentions in its opening chapters that different similarity measures have differing implicit assumptions, hinting at the existence of similarity dimensions. The Gene Ontology [62] defines two types of similarity measures, namely pairwise and groupwise, which are akin to kinds of similarity, albeit specific to that ontology only. Another work on semantic similarity [26], classifies existing methods for biomedical ontologies with respect to scope (what entities are taken into account), data source (edges, nodes or other) and metric (used algorithm). Authors of [26] observe that methods that use different metric and

data produce different results, but all claim to produce a similarity score that is "universal". To the best of our knowledge [6] was the first paper in which different kinds of similarity were the explicit focus, and were given the name "dimensions". Finally, let us also note that an implemented ontology matching system ASMOV [9] utilizes four (*lexical*, *relational*, *internal* and *extensional*) dimensions that are weighted and summed to obtain the final score.

It's important to note that, ideally, similarity dimensions should form an orthogonal partition of a "total" similarity. Since, as discussed previously, the distinction lies in the kind of data used, the available knowledge should be partitioned into subsets, one for each dimension. Under this characteristic, the similarity scores for each dimension would be independent. In practice, however, such clear division is not always possible (see, following sections).

In this context, let us now introduce selected dimensions and formalization of a general case of pairwise comparison of entities in an ontology based on description logic.

External and internal dimensions

From the point of view of the origin of data, similarity dimensions can be categorized into *external* and *internal* ones.

External dimensions. An *External* similarity dimension uses information from outside of the main knowledge base or ontology. In case of entities, *external* methods use a small (likely atomic) part of an entity description that serves as an identifier, to find information about it in external sources. A good candidate for such an identifier is a label of an entity, because it is available for all named entities and may be written in a natural language. A complex description is, in such case, simplified into a single term, so that it can be easily identified and searched for in outside sources. The assumption behind this operation is that the entity has a meaning outside of the original knowledge base, and this meaning is relevant to calculation of similarity.

For a pairwise comparison of entities this means that the used methods are actually independent of the DL formalism used to describe the entities. The information we use for scoring comes from an independent, external source (possibly having its own formalization). For English words a method commonly used in ontology alignment systems [9, 10, 11] is to utilize the English WordNet ontology and calculate similarity with a method specific to WordNet (e.g. feature method that works on synsets).

External methods have two inherent weaknesses, as they rely on: existence of a good term that describes each compared entity, and on the quality and relevance of the external data source.

The *External* dimension gives a perspective on how similarity of entities is viewed outside of known and specific context. The lack of this specific context causes disambiguation problems. For that reason *external* similarity scoring is best suited for entities that are general and relatively insensitive to context. For instance

it is useful when performing entity resolution for duplicate detection in data analysis, because often our data describes a broad range of items from multiple domains (e.g. items from a big online store). One of the simplest techniques of doing that is to convert entities into a canonical form, that serves as a representative and could be used as an *external* identifier.

Internal dimensions. *Internal* dimensions are those that make use of information either explicitly provided in the knowledge base of compared entities, or inferred from that knowledge base. In this case we are not interested in any independent outside sources and assume that any knowledge we might use must come from what we already have in the knowledge base, or an ontology connected to it (e.g. via Linked Data [68]). Vast majority of similarity methods are *internal*; especially, ad-hoc ones that are restricted to a single ontology (e.g. [5] that works on WordNet only). Here, we assume that any discussed ontology can be expressed in description logic.

Usually, in description logic, entities compared *internally* are of the same type, i.e. both come from the same "box"—ABox, TBox or RBox (we compare concept with concept, role with role, etc.), although it is possible to compare "across boxes" (e.g. a concept with an individual), which is explained in what follows.

Note that some similarity dimensions have interpretations for both categories—*internal* and *external*, while others are exclusive to one category.

Lexical dimension

Lexical methods utilize dictionaries and lexical ontologies to asses similarity of entities (e.g., see [5]). In a general case of an ontology, *lexical* methods are *external*. Entities are considered in the context of a dictionary (where they are referenced to by an identifier) and not the original ontology. A pair of labels, or entity names, written in a natural language can be subject to the *lexical* dimension similarity methods. The methods themselves might be very complex and utilize big ontologies (such as WordNet).

Lexical dimension is most useful when entities have uniquely identifying labels. For that reason we can expect that, for example, comparing terms "dolphin" and "porpoise" will yield useful results. A simple *lexical* method for concepts could, for instance, extract the labels of entities and use WordNets *synsets* of the labels as features, in a feature-based method. On the other hand, this is not the case when labels are human names (e.g. Mary, Adam), because, even though technically being labels, those are a properties of an individual, rather than unique identifiers. In different ontologies these might refer to different people. Similar problem arises when identifying terms are words with multiple meanings (e.g. "seal"). Generally, any identifier that is sensitive to context of a knowledge base (like human names) is not a good candidate for a *lexical* similarity scoring. This is because in any external similarity dimension we lose the original context. In a *lexical* method the additional context we need to consider is the natural language itself (e.g. English,

French etc.). *Lexical* scores might differ between languages, because of varying sets of homonyms and many natural differences between languages. Despite this, as mentioned before, many ontology alignment methods use external *lexical* similarity as a way to find connections between ontologies that have no links defined between them. Sometimes the *lexical* scores are used as a bootstrap to discover other connections between ontological entities and improve the alignment.

Informally, the *lexical* dimension specifies similarity of names of entities in a dictionary. Unfortunately, it suffers from the problem common in dictionaries, i.e. ambiguity. So-called "word sense" disambiguation is a big issue in text processing [27] and semantic ontologies (e.g. applied to named entities [28]). Ambiguity of language negatively impacts accuracy of the *lexical* similarity score. Notice that, in the case of a well defined ontology, there is no ambiguity problem, because the entity descriptions are compared directly. When comparing terms we first need to find out what entity each term represents (what is the underlying entity) and then compare the entities. Miller's benchmark [43], often used to evaluate WordNet methods, does not, unfortunately, have explicit concept descriptions, and the word sense in each word pair needs to be decided solely on the two words in each pair.

In short, the interpretation of the *lexical* dimension is that entities *lexically* similar have names that are similar, according to a dictionary. In order for a *lexical* method to be semantic, it should not rely on any edit distance.

Co-occurrence

Another group of methods dealing mostly with the *external* dimension are the *co-occurrence* methods. Like *lexical* methods, they also use a single term or label (identifier). Similarity is calculated based on a highly controversial assumption that entities that often appear together are similar. For instance, the web search co-occurrence methods measure the number of web pages that contain both identifiers (or terms). Methods in this dimension are often used for text similarity scoring, and work under the assumption that words that appear together in a high number of text corpora are similar. More advanced co-occurrence methods distinguish between different meanings of words [29]. Their authors, realized that, like in the case of *lexical* methods, disambiguation is an issue. In data analysis *co-occurrence* is used as an evidence of similarity (called "linkage pattern").

Co-occurrence methods usually do not take into account the reason for two entities appearing together. For instance, they do not take into account that co-occurrence might be a result of a single event, local culture, specific names (e.g. names of sports teams), or even a coincidence. In this way, *co-occurrence* is an evidence of relatedness, but not necessarily similarity. Overall, *co-occurrence* methods are known to give questionable results [30].

Co-occurrence dimension is *external*, because it uses many data sources (e.g. web pages, documents etc.). A commonly used sources are those that are publicly available, such as Wikipedia [44], or Freebase [91]. Although, in an ontology we

might construct a co-occurrence method based on an assumption that entities that appear in a high number of axioms together are similar. Such methods would give results that would come under the same questions as ones from other co-occurrence methods [30]. Moreover, since the axioms contain detailed knowledge about semantic relationships between entities it is better to consider *why* the entities appear together, rather than disregarding that information. For this reason the type of axiom (e.g. RDF predicate, if available) should always play an important role in similarity scoring.

Interpretation of a *co-occurrence* similarity is, simply put, that entities often appear together and are referenced in the same contexts.

Taxonomic (sort) dimension

Similarity in the *sort* dimension (also called *hierarchical* or *taxonomic*) describes how similar entities are, according to data from taxonomy and, therefore, uses mostly the TBox.

Theoretically *sort* dimension is most easily described in terms of *types* of concepts (i.e. subsumers or ancestors). For instance, in a hypothetical ontology of genetic ancestry two classes of creatures are similar, if both are reptiles (they share a type). Similarity increases with each type that the creatures have in common. At the same time it decreases with each disparate type (e.g. when one creature is a lizard, and the other a snake). This is in accordance with the general tenets of the concept of similarity (see, section 1.3).

Practically, taxonomy is often visualized as a graph, where nodes are concepts and edges are IS-A relations. Because of the structure of a description logic taxonomy, each common type of compared entities lies on some path from the root ($\top$) to either of the entities. More precisely the commonality is defined by any path to the lowest common subsumer (LCS) of both entities. Any edge on such path is between two common types. Any edges from $\top$ to any of the entities that does not lie between $\top$ and LCS is an evidence of dissimilarity.

Many edge-counting methods (that use taxonomic ancestry of entities, [64]), some IC methods (like [55] or [81]) and feature methods (e.g. [31]) can be used in this dimension.

Formally *hierarchical* dimension includes information exactly about DL relations of subsumption ($\sqsupseteq$) and, consequently, inclusion ($\sqsubseteq$) and equivalence ($\equiv$). Recall that a concept is a specialization of all its types (classes), including the root, and a generalization of all its children (subclasses). The root ($\top$) is a generalization of any concept. Similarity measures that work on subsumptions usually take into account subsumers of measured classes, rather than descendants. For instance edge-counting methods "count" classes (types) that are on a path between the LCS and the root. Some IC methods make use of number of descendants (subsumed classes) to calculate "probability" of a node. *Taxonomic* similarity is also linked to distance between measured entities, either directly, or through IC of LCS.

Other than concepts, *sort* dimension can also be applied to roles or individuals. In some profiles of description logic roles have their own hierarchy (e.g. $\mathscr{H}$ subprofile of DL [65]) with a separate set of IS-A relations (whose domain and range are roles, not concepts) that also form a set of data for the *sort* dimension. In practice, however, the hierarchies of roles are almost never rich and deep enough to provide enough information for a useful hierarchical similarity score. Simply put, such score would not be useful. Individuals do not form their own hierarchy, however, there are ways to relate entities of that type to the taxonomy, e.g. the most specific class (MSC, [56]). Another method is to use only concept membership (asserted and inferred) for compared individuals as *taxonomic* knowledge. In this method we essentially compare sets of types of each compared individual, which is a good fit for a feature method, where each type would represent a feature. Since we can construct a set of ancestors for a concept and set of types for an individual (both ancestors and types are concepts themselves) it is possible to compare concepts with individuals using a feature method. Thus, in *taxonomic* dimension we can compare pairs of concepts, individuals and roles as well as a concept—individual pairs.

In $\mathscr{EL}$, concepts can be represented as an intersection of terms e.g. $C \equiv D_1 \sqcap D_2 \sqcap \exists p_1 \sqcap \exists p_2$ and $E \sqsubseteq D_1 \sqcap \exists p_3 \sqcap \exists p_4$. In this example, the knowledge that pertains to *sort* dimension is the part of the expressions that contains concept names, namely $C \equiv D_1 \sqcap D_2$ and $E \sqsubseteq D_1$. The role assertions are not considered a part of this dimension, so we do not take them into account. If we are interested in comparing the two example concepts C and E with respect to subsumption (note that $A \equiv B$ is equivalent to $A \sqsubseteq B$ and $B \sqsubseteq A$) we would use two expressions: $C \sqsubseteq D_1 \sqcap D_2$ and $E \sqsubseteq D_1$ that can easily be converted into sets (through itemization with respect to intersection) $[D_1, D_2]$ and $[D_1]$ respectively, and used in a feature method. For individuals a, b, assuming $C(a)$, $E(b)$, *sort* similarity of a and b is equal to *sort* similarity of C and E.

The general idea of "truncating" a complex description to one containing only symbols for concepts and constructors (to "extract" *sort* similarity) holds for more expressive DLs. For instance the expression $C_2 \equiv D_1 \sqcup \forall p_1.(D_2 \sqcap \exists p_1.D_2)$ does not seem to be easily subjected to "extraction" of *sort* terms. In practice, however, we can rely on semantic reasoners to build an inferred taxonomy that puts all named classes in order with respect to subsumption (and inclusion) while taking into account complex expressions [40]. New concepts, such as MSC, can also be put in a proper place in a taxonomy with the help of semantic reasoners. It is also common for the taxonomy tree to be explicitly created (asserted) by the author of the knowledge base. In *sort* dimension we are only interested in the existence of IS-A relation between entities and not the reasons for existence of such relation. Combining asserted and inferred hierarchies produces data that accurately represents *taxonomic* dimension.

In summary, *taxonomic* similarity of two entities is interpreted as the entities being of similar type or class, or sharing a number of types. While, in layman terms a "type" is a vague term, it has a very specific meaning in practical applications i.e. ontologies.

Descriptive dimension

From a theoretical point of view the *descriptive* dimension contains properties that an entity "has", as opposed to what it "is" (which is covered in *taxonomic* dimension). For animals, similarities in size, weight or age belong to the *descriptive* dimension. Generally speaking, *descriptive* dimension encapsulates attributes, characteristics, or properties of entities. Properties such as "having a child" are also included (and distinct from "being a child", which belongs to *taxonomic* dimension). Again, the more disparate attributes, the less similarity and vice-versa.

In certain ontologies, clearly distinguishing between *taxonomic* and *descriptive* data might be problematic, when it comes to entities that form a hierarchy. The difference between two dimensions, and whether they overlap or are entirely orthogonal, comes down to the way the hierarchy is constructed by the authoring ontology engineer. Let's consider a hierarchy of classes. A taxonomy might be created in an entirely expert-driven fashion, in which case it would not contain any explanation as to why any given class has the subsumers that it does. It would be simply an assertion of expert knowledge, stating that any instance of an example class *A is* of every type that subsumes *A*. On the other hand the taxonomy construction might be driven by roles of every class. Here, the reason for enclosing two classes in a subsumption relation is that they share a role restriction. In this case, a subsumption implies that *A has* a property that is shared among all its subsumers. In other words the basis for subsumption is inheritance of role restrictions. Informally, if an information about a role is "included" in a taxonomy (or used in its creation), then it overlaps with the *taxonomic* dimension, where it is included implicitly. The *descriptive* dimension considers all roles explicitly. More formally, orthogonality of *taxonomic* and *descriptive* dimensions depends on whether the ontology follows the principle of cognitive saliency [98]. Overall, this principle states that new concepts are created and subsumed only when there is a need to differentiate them, and put them in their own class. This principle is, often unknowingly followed in a lot of ontologies, and one can assume that the *taxonomic* and *descriptive* overlap is small or does not exist at all.

Practically, in ontologies that have both subsumption relations and role restrictions, the taxonomy includes results of both methods described above—expert assertion and inheritance. Specifically the inherited roles are the cause of partial overlap between *taxonomic* and *descriptive* dimensions. Notice that for any two concepts, the set of role restrictions that they have in common is at least the set of roles of their LCS, because both concepts inherit those roles from the LCS. In a very special case, where each class has only one non-inherited role restriction, each IS-A relation has a corresponding role restriction. Numerically, this means that number of contributing relations for both dimensions is exactly the same, so we can expect the results from both dimensions to be close. Such cases are very rare in practice, where some roles are the explicit reason for the shape of taxonomy, and some are independent of it. In an example biological ontology of creatures, properties such as type of reproductive system are the base of phylogenetic taxonomy. Other, such as diet or geo-spatial distribution are not considered in phylogeny. This is because

they are not *inherited* genetically, which is the basic requirement for a phylogenetic subsumption.

Notice that, even when considering full set of roles of a concept (both inherited and not inherited), its cardinality might be different from the cardinality of the set of types (ancestors). For every item in the set of types we might have any number of role restrictions inherited for that type. In other words, every ancestor contributes only one piece of data (one superclass for *taxonomic* dimension), while roles inherited from the ancestor might contribute (to *descriptive* dimension) a different number (0 or more). For two concepts with complex descriptions that are on the first level of taxonomy (i.e. direct children of $\top$), if we apply an edge-counting method, their descriptions are essentially irrelevant for *taxon omic* similarity score (only distance to each other or $\top$—their LCS—matters). In fact, this is the case in the *semantic sensor network ontology* (SSN) [69], where concepts close to $\top$ have a lot of roles. Those roles could have big impact on a similarity score, but are disregarded by *taxonomic* methods. This observation suggests that a clear way to distinguish the two discussed dimensions is to consider only non-inherited part of complex concept description, or, alternatively, not consider part of the description inherited from the LCS. The lack (in practice) of full overlap of dimensions suggests that *descriptive* similarity is useful along side of *sort* similarity and produces results with a different interpretation.

More formally, in *descriptive* dimension we are interested in relations that are roles and are not of type IS-A. Hierarchy constructed from any such relation is not taken into account. Instead, only existence of a relation and its value is considered. In DL terms, those are either role assertions (e.g. $r(a,b)$) in case of individuals, or quantified restrictions (e.g. $\exists p.C$, $\forall t.5$) in case of concept descriptions. *Descriptive* dimension fits naturally with feature methods, because we can treat each role assertion or restriction (a "descriptive" expression) as an item in a set of features, either for TBox or ABox. In $\mathscr{EL}$, extracting a set of such features from concept description is simple and very similar to the method described for *sort* dimension (it yields sets $[\exists p_1, \exists p_2]$ and $[\exists p_3, \exists p_4]$ for concepts C and E, defined earlier, respectively).

Individuals do not form a hierarchy, so comparisons between this type of entities do not suffer from overlap with *taxonomic* dimension and as such are a good fit for *descriptive* dimension. For a set of statements (role assertions) $r(a,b)$, $r(b,b)$, $r(a,c)$, $p(a,c)$, $t(b,5)$ about individuals a, b and roles r, p and t, the first two ($r(a,b)$, $r(b,b)$) contribute to similarity of a and b, because the predicate (role) and object (individual) are the same for both a and b. Expressions $r(a,c)$, $p(a,c)$ and $t(b,5)$ contribute to dissimilarity of a and b, because those assertions do not share both role and object for a subject of a or b.

Comprehensive implementation of *descriptive* dimension in expressive DLs is highly problematic. While it's relatively easy to construct a transformation of a complex description to a normal form (e.g. conjunctive normal form) there is no universal way to compare complex restrictions. For instance there is no universally accepted method to calculate similarity of each pair of $\exists r.A$, $\forall r.A$ and $\exists r.B$ other than to treat those as entirely different (similarity score of 0), even though intuitively we might conclude that, since all 3 expressions pertain to the same role r,

they are not *absolutely* different and the similarity score should not be zero, even if its close to it. Unfortunately, when it comes to roles in DL, the canonical approach is that they can be either identical or not, with no degrees of similarity in-between. The binary treatment of role restrictions or, widely speaking, features is a big weakness of many similarity methods. Moreover, comparison of complex descriptions, especially in expressive DLs, is a complicated problem, and is beyond the scope of this paper. For those interested, [80] proposes a method of comparison of complex descriptions.

It should be stressed that, unfortunately, existing methods usually do not distinguish between *taxonomic* and *descriptive* data, instead implicitly assuming that every role restriction contributes to a concepts place in a taxonomy and has no additional bearing on similarity. Consequently, there are no methods known to us that are purely *descriptive*. That being said, feature set methods are a natural fit for this dimension, because of the clear division between *descriptive* features, and other features.

Descriptive similarity is (informally) interpreted as standing for similarity of properties, attributes or characteristics, i.e. the items that describe what an entity "has".

Other dimensions

Up to this point we have presented four similarity dimensions that are very general and thus widely applicable. There are many other ways to divide knowledge, as was suggested in section 1.5. Each of the relevant works [12, 22, 49, 5, 6, 26] uses different kind of semantic relations and axioms. One could even argue that any partition of knowledge forms a set of semantic dimensions. The ones proposed in this paper were designed (on the basis of analysis of existing methods and ontologies) to be relatively simple in interpretation and generic enough to be available in almost any knowledge base. There are, however, other, more specific dimensions, that are worth mentioning.

Let us start from the the *membership* dimension. It can be used to measure similarity (only) between concepts by gathering and comparing sets of individuals that are of specific type. Compared to others dimensions, this one produces simple data even for expressive DLs, because the membership function is a binary predicate—an individual either is or is not of a given type. From simple statements $A(a), A(b), C(a)$ we know that concept A has members a and b, and C has member a. This knowledge can be easily used to construct a feature method. The *membership* dimension is implicitly used in [56] where authors build feature sets composed of members and calculate similarity in a way very similar to Tversky's feature method. Because any individual of any type A is also of all types that are ancestors of A, the *membership* dimension uses data that overlays, in part, with the *taxonomic* dimension, but still brings its own perspective on similarity.

Separately, the *descriptive* dimension contains knowledge about all of the properties without discrimination. One simple way to create a new similarity dimension is to isolate a set of types of roles from the *descriptive* dimension. The resulting set should have its own specific interpretation to be considered a separate dimension.

An example resulting from this method is the *compositional* dimension. It is comprised of roles that denote "being a part of," "having parts," "having ingredients," etc. It has a very clear interpretation and, as humans, we can often look at composition of any physical object. Formally, it is represented by roles such as *hasPart*, *isPartOf*, *isIngredient*, etc. In SSN [69] this kind of relations are represented by the *hasPart* role (inherited from *DUL* ontology). A similar role exists in WordNet [61] (also named *hasPart*) and in many other ontologies.

Another "sub-descriptive" dimension is the *physical* dimension. It contains all roles that describe any kind of physical characteristic. What roles are included specifically varies between ontologies. They might include size (e.g. height, width, area), mass, color, shape and others.

A practical problem with the subdivision of the *descriptive* dimension is that application of a dimension constructed by this method requires specific roles. Even guided by the interpretation, the specific dimensions might be represented by different roles in different knowledge bases. In one ontology the *physical* dimension would include a *hasWeight* and *hasHeight* roles, while in another by a *hasArea* role. A third ontology might not contain any roles relevant to the *physical* dimensions and, therefore, the *physical* similarity score would not be available. Any subdivision of the *descriptive* dimension generally means a loss of universality, i.e. one cannot apply our new dimension to every ontology. Another downside of this method is that the "sub-descriptive" knowledge in a very obvious way overlaps with the *descriptive* dimension. As a consequence, for instance, the *compositional* score and *descriptive* score are not independent (in fact, one is contained within the other), and the dimensions are not orthogonal. On the other hand, sub-descriptive dimensions are easy to implement in edge methods, such as [91]. What is required is simply to use only edges of a certain type, instead of all edges. One needs to be mindful that not all edge types appear often enough in a graph to form an interesting and useful dimension.

Let us recall that conversion of roles into a set of features is easy for simple DLs, but gets complicated for more expressive DLs. This is relevant for the *descriptive* dimension and its sub-dimensions, where we need to compare DL expressions (role assertions or restrictions). For a sub-dimension that contains only role p, a simple, single-term expression, such as $p.D_1$ is easy to parse and compare. A complicated expression, such as $p.((D_1 \sqcup (\forall p.D_2)) \sqcap (D_1 \sqcup \forall p.(\exists p.D_2)))$ is difficult to use in a comparison with others, because the class expression under the property restriction in the example is very complex. Moreover, it might have many equivalent forms, which are relevant in practical implementations of similarity algorithms. The simplest approach to solving this problem is to consider only the binary similarity of complex expressions.

Section 1.8 contains a description of interesting properties of semantic similarity dimensions that should be considered when designing new dimensions. Before that, let us present an example of application of dimensions introduced up to this point.

1.6 Example of multi-dimensional similarity

Let us consider an example of dimensional similarity scores in a mock-up biological ontology. The ontology in question (see Fig. 1.1) is an extract of a phylogenetic ontology with added roles. It compares three concepts—short-beaked common dolphin [70], silvertip shark [71] and lesser electric ray [74] denoted D, S and R respectively. Taxonomy describes the current understanding of the genetic ancestry of these creatures. It is complemented by roles selected to best aid in presentation of the idea of semantic similarity dimensions. The roles represent traits or features that are not genetically inherited and, therefore, in the example the *descriptive* dimension does not overlap with the *taxonomic* one (see, section 1.5). Note that this is in no way a complete set of information about these creatures. Data contained in these roles comes from [70, 71, 72, 74, 75] and was prepared with ease of understanding of the example in mind. Note that biology is not the main focus of this paper and accuracy of the data was not verified. This example is meant to demonstrate usage and indicate usefulness of similarity dimensions. Let us note that all used formulas are symmetric, normalized and have the properties of minimality and maximality.

Data used by *taxonomic* methods is the hierarchy of concepts and in this example there are 20 phylogenetic concepts (classes, including $\top$). Resnik's method [50] specifies similarity as the IC (information content) of the MICA (most informative common ancestor), which in the example is equivalent to the LCS (least common subsumer); $S_{Res}(X,Y) = IC(MICA(X,Y))$, $IC(e) = -\log(p(e))$. According to this method, similarity scores are as follows: $S_{Res}(D,S) = IC(CHORDATA) = -\log(\frac{18}{20}) \approx 0.105$, $S_{Res}(S,R) = IC(SELACHIMORPHA) = -\log(\frac{9}{20}) \approx 0.799$, $S_{Res}(D,R) = IC(CHORDATA) \approx 0.105$. Note that this example contains only a fraction of available phylogenetic classes and in a full ontology Resnik's method would give a different score, because IC is sensitive to the total number of concepts, which is the basis of calculating the "probability" of a concept. Calculation of Jaccard index $J(A,B) = \frac{|A_f \cap B_f|}{|A_f \cup B_f|}$, where A_f is a set of features of A and assuming that each ancestor of a concept (including $\top$) is a feature, gives the following results: $J(D,S) = \frac{3}{16} \approx 0.188$, $J(S,R) = \frac{6}{14} \approx 0.429$, $J(D,R) = \frac{3}{16} \approx 0.188$.

In the *descriptive* dimension, the data we use are role restrictions. We can, again, use Jaccard index, this time using roles as features. This is an indication that we can use one method to calculate similarity in many different dimensions. In this example D and S have 6 roles each, while R has 4 roles. As mentioned in section 1.5, the simplest way to compare two values of a single role is to say that the similarity is binary (1 only if those values are identical and 0 otherwise); i.e. $Sim(r.5)$ and $Sim(r.4.99)$ is 0, despite their perceived numerical "closeness." Under this condition,

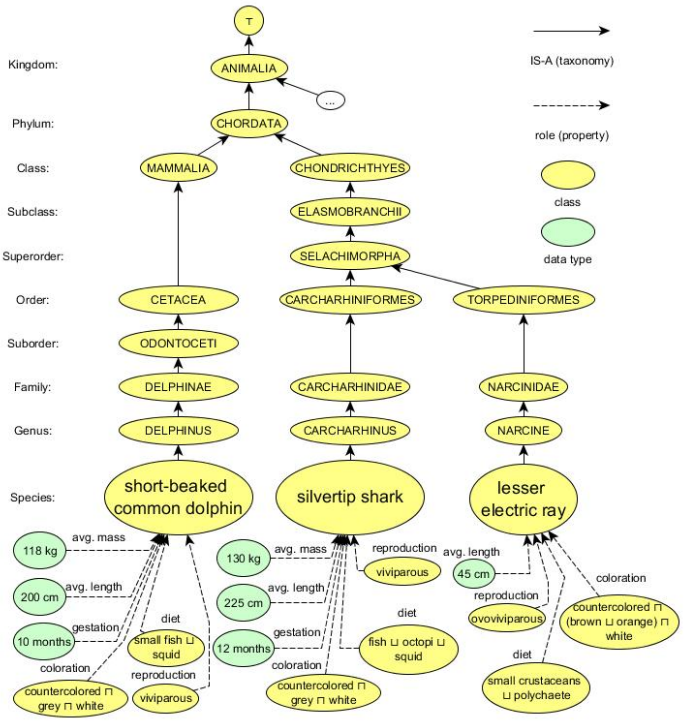

Fig. 1.1 Phylogeny example

descriptive Jaccard scores are as follows: $J(D,S) = \frac{2}{10} = 0.2$, $J(S,R) = \frac{0}{10} = 0$, $J(D,R) = \frac{0}{10} = 0$.

Final dimension considered here is the *physical* dimension that is meant to represent any physical feature, i.e. roles for mass, length and coloration. In order to better represent difference between numerical values, a simple ratio method is used for data values of the same role (assuming the same unit). This similarity is equal to the smaller value divided by the larger one $S_{val}(k^r, l^r) = \frac{\min(k^r, l^r)}{\max(k^r, l^r)}$, where k^r and l^r are values of role restrictions or assertions, about the same role r. For instance similarity of average weight between D and S is $\frac{118}{130} \approx 0.907$. Total similarity is this dimension is calculated by taking arithmetic average over similarity of each relevant role. The scores are: $Sim^{avg}_{ph}(D,S) = \frac{\frac{118}{130} + \frac{200}{225} + 1}{3} \approx 0.932$, $Sim^{avg}_{ph}(S,R) = \frac{\frac{45}{225} + 0 + 0}{3} \approx 0.067$, $Sim^{avg}_{ph}(D,R) = \frac{\frac{45}{200} + 0 + 0}{3} \approx 0.075$. Using the same method of simple arithmetic average the results for the entire *descriptive* dimension are as follows: $Sim^{avg}_{desc}(D,S) = \frac{\frac{118}{130} + \frac{200}{225} + \frac{10}{12} + \frac{1}{4} + 1 + 1}{6} \approx 0.813$, $Sim^{avg}_{desc}(S,R) = \frac{0 + \frac{45}{225} + 0 + 0 + \frac{2}{4} + 0}{6} \approx 0.117$, $Sim^{avg}_{desc}(D,R) = \frac{0 + \frac{45}{200} + 0 + 0 + \frac{2}{4} + 0}{6} \approx 0.121$.

Table 1.2 Approximate similarity scores

	$Sim(D,S)$	$Sim(S,R)$	$Sim(D,R)$
taxonomic			
Resnik	0.105	0.799	0.105
Jaccard	0.188	0.429	0.188
descriptive			
Jaccard	0.2	0.0	0.0
arithmetic average	0.813	0.117	0.121
physical subdimension			
arithmetic average	0.932	0.067	0.075

Analysis of results

Obtained similarity scores are summarized in Table 1.2. Observe that *each method produces different similarity scores*, even in the same dimension. In the *taxonomic* dimension, Resnik's and Jaccard's methods produce different scores. This is, for instance, because of the assumption of Resnik that distance to the root in an ontology (a level) is significant. The levels of example concepts do not correspond with levels of phylogenetic classification, e.g. the dolphin does not have a biological *subclass* or *superorder*, so technically its *order* (CETACEA) is on the same ontological level as *subclass* of the shark (ELASMOBRANCHII), even though intuitively (and in accordance with biological research) an *order* should be more informative than a *subclass*. Such structure is a good example of a graph, in which edges do not uniformly represent the same value of difference in specificity (this was described in more detail in section 1.4).

Differences between dimensions are very apparent in the results. In particular S and R have very small *descriptive* similarity (Jaccard gives a score of 0), while their *taxonomic* similarity is significant. Explanation of those results lays in the fact that *descriptive* features from the example were not used when constructing phylogeny. Features such as diet, type of reproduction, coloration, period of gestation, and others vary in the same *genus*, so *species* are not classified based on those characteristics. Purely *taxonomic* methods (such as Resnik's) do not take such features into account at all. Consequently, in this case, *descriptive* results are independent of taxonomy.

Another noteworthy observation is that the *physical* dimension score does not coincide with the *descriptive* Jaccard score, even though the former is, theoretically, a subdimension of the latter. This difference stems from difference in used methods. The *physical* arithmetic average method takes into account degree of difference between values corresponding to the same role, while the *descriptive* one does not, and only accepts identical values as similar. This, very simple, method works for this example, but cannot be applied universally (e.g. because of the division by zero problem). Unfortunately, disregarding custom ad-hoc methods (that work well, but cannot be easily applied outside of one specific ontology), there is no good and universal method that would compare complex descriptions in expressive DLs in an in-depth manner.

Notice that ordering of similarity changes between dimensions. *Taxonomicaly S* is closer to *R* than to *D*, while *descriptively S* is closer to *D*. Taking into account interpretation of the used dimensions this suggests that short beaked dolphins and silvertip sharks look similar (high *physical* similarity), but their evolutionary ancestry is different (low or average *taxonomic* score). This statement is possible because separate dimensions of similarity have been independently evaluated and thus can be interpreted on the basis of their own semantics. It is impossible to infer such information from a single score.

One dimension that was not included in the example (for sake of brevity) is the *compositional* dimension. It would comprise of physical "components" of the animals with additional details, for instance, fins (e.g. small pointed dorsal fin), details of bone structure (e.g. serrated teeth), specific organs and functions (e.g. Ampullae of Lorenzini [73]). In this case the *hasPart* properties would refer to body parts. Note that this dimension has an interpretation in the context of phylogeny that fits the general interpretation and description from section 1.5.

The final answer to the question of "how similar are two concepts?," for the dolphin and shark, according to Resnik's method is 0.105. According to the method of dimensional similarity *D* and *S* have *taxonomic* similarity of 0.188, *descriptive* similarity of 0.2 and *physical* similarity of 0.932. However, on the basis of the discussion presented thus far we strongly argue that dimensional scores are much more informative and, thus, useful to the "end user." We present further justification of this statement in section 1.10. This being the case we propose a dimensional method, which produces multiple scores that can be organized into a *dimensional similarity vector*. Let us discuss this now this idea in some detail.

1.7 Combining similarity dimensions

As discussed so far, the canonical approach to similarity scoring is to present a single number as a result. Sometimes a range of intermediate results is calculated, in which case a method of combining those results into one, such as a weighted sum, is utilized. In this case, there are many weighting methods including metrics [77, 76], machine learning [5], aggregation operators [78, 79] and others [23].

For instance, in [5] authors used a weighted sum of 5 similarity scores of WordNet concepts and various machine learning methods. The weights were trained against a (human) survey similarity scores for pairs of concepts. The authors remarked that for the scores from each dimension considered alone (for a test set of 20 pairs), each dimension at least once (i.e. for a specific pair) provided the best score (i.e. closest to training data). This led to the conclusion that the trained weights, even though useful for this specific application, may not be a good fit for a different domain or ontology. Nevertheless, according to [5], results from multiple dimensions are more useful than any individual dimension used separately.

Another work [9], calculated the score as a weighted sum of 4 intermediate (dimensional) scores. The intermediate scores were not deemed to be individually rele-

vant and were only considered as parts of the weighted sum. Unfortunately, although the authors claim that the weights were "determined experimentally", the specific method of choosing weights was not described. The authors also cite problems with choosing good fixed weights [82], some of which are reiterated below.

The advantage of weighted sum is that the final score includes (and combines) a very broad range (possibly all) of available knowledge. Good set of weights offsets the possibility of overlap of dimensions by adjusting overlapping scores. A disadvantage is that there is no indication that weights calculated for one ontology give good results for a different one and recalculation of weights is expensive and requires a good training set (which may not be easy to deliver). Using a predefined metric or operator is less computationally intensive, but suffers from accuracy problems [23]. Overall, to the best of our knowledge, there is no weighing method that would produce a "universal" set of weights. Universality of weights, in the context of similarity scoring, means that one set of weights produces "good" (i.e. as compared to benchmarks) results for any testing set. So far no good method (training or otherwise) applicable to a wide range of ontologies and problems was found. This leads to a conclusion that it is extremely likely that weights are problem-specific, or ontology-specific.

In contrast, the dimensional similarity method proposed here produces a single score for each dimension. Those scores may be presented in a structure of the *dimensional similarity vector*, in which each cell contains a score from one dimension. While such vector can be then weighted and "reduced" to a single number, in the proposed approach the vector itself, as a whole, is to be used as the result.

Even though presenting the dimensional similarity vector as the final score goes against the established methods, it has clear advantages. First, we avoid the aforementioned problems with finding a good set of weights, which is very significant since the existing research suggests there might not be a good universal one. The gain is the amount of information that is contained within each vector cell. As mentioned, in section 1.5, the interpretation of a similarity dimension is helpful when designing dimensional algorithms, but it also provides useful information about the final dimensional score. Understanding what each score stands for is helpful when deciding what knowledge is relevant to our particular problem. In a sense, it is an avoidance of the universal weights problem, because, assuming that weights represent importance, we don't consider weighting a part of similarity scoring. Instead, the implicit "weighting" is done after scoring in each dimension, when we apply the results to solving a specific problem. From this perspective, we are free to use (or disregard) data from any subset of cells from the full similarity vector. Guided by the interpretation of similarity dimensions we can decide what dimensions of similarity are useful in the context of the problem that is being addressed.

Good understanding of dimensions also helps with correct interpretation of overlap(s) between them. In case of a single final score it is impossible to, for example, "subtract" the impact of a taxonomy, in cases where we are not interested in this dimension of similarity. Moreover, since usually the weights are hidden from the user, it is not possible to know the impact brought about by new data introduced into an

ontology, without either experimentation, or analysis of the code (i.e. reverse engineering), or documentation of used algorithm.

For the weighted sum there is also a general question—what is the actual meaning/interpretation of applied weights? If we assume the (intuitive) understanding that weights represent importance of dimensions, we may conclude that in a survey benchmark (like the Miller's one [43]) dimensions had some given importance to the participants. This approach, however, is problematic when it comes to automatic methods, because there usually is more than one set of weights that can produce the same weighted sum, for a single pair of entities. The hypothetical "importance" weights might also change on a pair by pair basis. Since the weights might not be unique, we cannot decidedly say that they represent importance, as viewed by the survey participants. Using a subset selected from cells of a dimensional similarity vector we essentially make a degenerated ad-hoc weighting. What we mean by that, is that choosing that we want only taxonomic score is equivalent to setting the weight of taxonomic score to 1, and rest to 0. For a subset of 3 dimensions, each weight for the chosen ones would be $\frac{1}{3}$, zero for the rest, and so on.

Example revisited

To visualize the problem let us propose a few sets of weights for the example from section 1.6. Table 1.3 describes an example with three hypothetical sets of weights w_1, w_2, w_3. The weights are used to obtain a single similarity score for two cases—comparison of Shark with Dolphin ($Sim_{Total}(S,D)$) and Shark with Ray ($Sim_{Total}(S,R)$). The weighted sum is made from three dimensional scores—*taxonomic*, *descriptive* and *physical* (values calculated in section 1.6 are recalled in the top part of the table).

The first set of weights w_1 assigns approximately equal value to each dimension. The resulting "total" scores are 0.41 and 0.29 for $Sim_{Total}(S,D)$ and $Sim_{Total}(S,R)$ respectively. Second set of weights w_2 indicates that the taxonomic dimension is decidedly more important than the other two and results in the scores of 0.34 and 0.42. Lastly, w_3 is a set of weights trained so that $Sim_{Total}(S,D)$ and $Sim_{Total}(S,R)$ are close in value (the result is approximately 0.4 for both).

Table 1.3 Similarity dimension weights example

	Shark				
	Taxonomic	*Descriptive*	*Physical*		
Dolphin	0.105	0.2	0.932		
Ray	0.799	0.0	0.067		
				$Sim_{Total}(S,D)$	$Sim_{Total}(S,R)$
w_1	0.33	0.33	0.33	0.41	0.29
w_2	0.5	0.25	0.25	0.34	0.42
w_3	0.47	0.20	0.33	0.40	0.40

First, notice that for w_1, because of the actual dimensional scores, the *physical* dimension has the highest contribution (i.e. highest value) to similarity of Shark and Dolphin, while the *taxonomic* dimension is the strongest in Shark and Ray comparison. This is in no way apparent in any of the Sim_{Total} scores.

More importantly, the final score Sim_{Total} has a very vague interpretation for any set of weights. All we can say about those numbers is that, depending on the weights, similarity of Shark and Dolphin is either greater, smaller or equal to that of Shark and Ray. The w_3 case indicates that a Shark is just as similar to the Dolphin as to a Ray, while the other two cases each produce an ordering of the two similarities. Any of those results can be put into question, depending on the perspective. A layman would classify Dolphin as much closer to Shark simply because the Ray looks nothing like the other two creatures. An expert in biology would, however, see much more differences and similarities in all creatures and would give a different score. Finally, an expert working specifically in phylogenetics would say that (phylogenetically) Shark and Ray have more in common than Shark and Dolphin. Traditionally, in a survey, the results would be averaged and produce a number that none of the participants exactly agrees on, but is representative of the average opinion. We conjecture that in similarity scoring, since the "average opinion" does not represent any actual perspective, it has a diminished usability. None of the scores—for w_1, w_2, w_3, or average of those, agrees with any other and yet, since the methodology is formally correct we cannot say that any of them are wrong, unless we adopt a specific perspective, e.g. to solve a specific problem in biology. Moreover, even from a particular perspective it is not possible to learn from the final score what information was most important (i.e. what were the weights for each dimension), if we only look at Sim_{Total}. Knowing that none of the total scores is indicative of every of the hypothetical survey participants, led us to believe that modeling different perspectives requires different weights, and none of the weight sets is, in general case, "more correct" than the other.

The "total" weighted sum score is contrasted with the dimensional score. Here the result is the dimensional similarity vector with 3 cells, one for each considered dimension, e.g. $Sim_{Dim}(S,D) = [0.105, 0.2, 0.932]$ for a set of dimensions [*taxonomic*, *descriptive*, *physical*]. Separately, each cell contains explicitly a single number, but also (implicitly) an explanation of the score in the form of interpretation (or description) of the dimension. The information contained in the vector lets us discern different kinds of similarity and learn that, for instance, Shark and Dolphin have high *physical* likeness, but their genetic ancestry (*taxonomic* similarity) is low. We can afterward decide whether this similarity dimension is relevant to solving our specific problem, or, in other words, whether it fits our perspective. Notice that this is useful both to an expert, and a layman. The first will learn much more from information about similarity in genetic taxonomy, rather than overall similarity. The latter will find more understanding in information about physical similarity of creatures, rather than some vague "universal" similarity.

To summarize, our recommendation is that the single number score (Sim_{Total}) should be used whenever it is required by a methodology—e.g. as an input to another method that accepts single number only, or to compare results with benchmark data

(which usually gives only one number for any pair of entities). In other cases we recommend the use of the full similarity vector (Sim_{Dim}), or a selected subset of dimensional scores, especially when the similarity score is presented to a user (as opposed to an automated system) that has a specific problem to solve. Dimensional score, simply put, gives the user more information without (possibly overwhelming and gratuitous) technical details of implementation and algorithm structure.

1.8 Properties of dimensions

None of the relevant works [12, 22, 49, 5, 6, 26, 9] that present some form of division of knowledge (collectively labeled as *dimensions*), in the context of similarity scoring, gave any formal reasons to support their specific choice of dimensions. The division of knowledge in each of the works was guided by authors intuitions and was tailored to fit the needs of a particular implementation of a given similarity algorithm. In this section we outline characteristics of dimensions that may support a decision as to, which dimensions to use, as well as guidelines for creation of new dimensions.

In our approach to semantic similarity dimensions, the *meaning* (semantics) of the dimension is absolutely essential. Table 1.4 summarizes the informal meaning (i.e. interpretation) of dimensions described in section 1.5.

Table 1.4 Interpretation of dimensions

Dimension	Interpretation
Lexical	Entities are *lexically* similar, when the words used to label them (i.e. their names) are similar according to a dictionary.
Co-occurence	Objects are *co-occurrence* similar, when they often appear together.
Taxonomic	Objects are *taxonomicaly* similar, when they are of similar class, kind or type.
Descriptive	Objects are *descriptively* similar, when they have similar properties, attributes or characteristics
Physical	Objects are *physically* similar, when their physical characteristics and appearance is similar
Compositional	Objects are *compositionaly* similar, when they have similar set of parts or ingredients
Membership	Objects are *membership* similar, when they have similar sets of representatives, instances or members

Ideally, the meaning (semantics) of each dimensions should be easily understandable even to a layman. Note that each dimensions from table 1.4 can be summarized in a single sentence. Such concise summary on a high level of abstraction should be accompanied by a more verbose explanation. For instance the meaning of "type" in the *taxonomic* dimension summary is clear to an ontology engineer, but it might be confusing to others. It is crucial that the semantics of every dimension is properly explained. This is because the explanation of the meaning is the main guideline

when it comes to actual implementation of the similarity algorithm. As noted in the explanation of the *physical* dimension, we can expect that the structure, roles and even semantics of different ontologies will vary greatly. Despite this, the same similarity dimensions should be applicable to many ontologies. Even implementations of *taxonomic* similarity might differ considerably, especially when we extend our problem space to systems that place, or relax, specific restrictions on taxonomies (e.g. multiple inheritance, no common root, etc). The *idea* behind each dimension must be independent of any particular structure and should not make any unnecessary assumptions. It should have meaning outside of computer science and encompass many possible implementations within it.

A common, agreed upon, interpretation of a similarity dimension allows for direct comparisons of similarity scores from different methods. It also allows to distinguish that scores from different dimensions (e.g. *physical* and *compositional*) refer to a different kind of similarity and we can expect that they will not be related to each other, even for the same pair of objects. A "total" similarity score has only a very vague meaning of a "degree of similarity" and even though there is no basis for this, we expect such scores to be close to some idealized target similarity, and, therefore, close to each other. As is apparent from table 1.2 the scores (for the same ontology) vary depending on selected algorithm and data fed to it. This shows that there does not exist one universal and ideal similarity, outside of artificially constructed references (sometimes based on averages from a survey).

Even though the general understanding of what a given similarity dimension represents is always the same, its informativeness is improved when we put it in a context of a specific ontology. For instance, in case of example from section 1.6 the *taxonomic* score has an interpretation of *phylogenetic similarity* (in general terms, evolutionary ancestry) on top of the general one (given in table 1.4). In the example ontology the taxonomy contains exclusively classes of living organisms and the position of an entity in this taxonomy is representative of its position in evolutionary tree (phylogeny). Understanding of what phylogeny is and how it is constructed improves the understanding of this dimension even more. Note that there may be many phylogenetic ontologies, each with (slightly) different taxonomy. The general interpretation of the *taxonomic* dimension is the same for any ontology. The *phylogenetic* interpretation of this dimension is the same for any *phylogenetic* ontology. The details of a *very specific* interpretation of the dimensions may differ in different phylogenetic ontologies, but the general interpretation stays the same. A well-defined semantic similarity dimension should be interpretable on many levels. In other words, it should have the *granularity* that is most useful.

Granularity

Let us now consider the fact that the *granularity* of a dimension is directly related to how detailed and specific is the explanation of its interpretation. In other words *granularity* is the amount of information carried in a description of a dimension.

The least *granular* (or informative) notion is simply what we referred throughout this text as "universal similarity". The "universal similarity" is mostly understood as an intuitive concept and its meaning may be studied in the field of philosophy, not computer science. "Semantic similarity" is almost as vague of a term, describing the similarity of meaning. There is no formally strict definition of it and, although some define it as a metric. However, mathematical properties of semantic similarity are not set in stone, as mentioned in section 1.3.

To visualize the *granularity* of dimensions let us use a simple example in the context of the MusicBrainz database [99]. MusicBrainz contains data about worldwide music industry i.e. artists, albums, music companies, music genres etc. It is available in many forms, one of which is LinkedBrainz—a linked data version of the database. Usefulness of LinkedBrainz can be enhanced by exploiting the linked data and connecting it to dbPedia, which contains information that is directly related.

Music albums in MusicBrainz are "releases" (*mb:release*) of type "album" (*mb:album*). A descriptive similarity dimension for MusicBrainz is more informative than just "semantic similarity" and it is applicable to any concept within MusicBrainz in the same way as in any other knowledge base. For a *mb:release*, it denotes similarity of all its properties such as *mb:artist*, *mb:title*, *mb:label*, *mb:format* and others. In particular the *mb:type* is not included here. In simplistic terms the gain of information stems from restricting the fields that we include in the similarity scoring to a smaller set. This is also true if we design a similarity dimension for any specific ontology. Doing that, however, we loose the ability to directly apply our new dimension to any other ontology. An increase of specificity (information) means a decreased range of possible applications. This is particularly apparent when the description of a dimension specifically mentions a property. For instance, in order to group albums by musical era we need to know the similarity of their release date. Such "album-time-of-release" dimension is very specific, because it can only be applied to an ontology that describes music albums and stores release time data. It is also very informative—we know exactly what data is used and, since time data is numerical, we can directly relate it to a syntactic similarity, or closeness of numbers. The possible data and algorithms used in this dimension are very restricted. Separating the data in such dimension does not bring any immediately apparent benefit and is, frankly, not necessary or advised. This is in stark contrast to low granularity dimensions, e.g. *descriptive* similarity. Dimensions of moderate similarity are an attempt to strike a balance. For instance, the *compositional* dimension is only applicable to ontologies with appropriate roles (e.g. *hasPart*), but since many ontologies do in fact have such roles, this requirement is not very restrictive. The granularities of this example are summarized in table 1.5.

In summary, a good design of similarity dimensions exhibits a balance between informativeness and applicability. From the point of view of *granularity*, similarity dimensions can be put on a spectrum between very specific syntactic similarity and very vague (semantic) similarity. Low informativeness gives a wide range of possibilities when it comes to implementation. High granularity leaves no doubt when it comes to the meaning of such highly granular, dimensional similarity score. The

Table 1.5 Information in similarity

Relative Informativeness	Similarity description	Similarity interpretation
0	Similarity	Likeness, closeness
1	Semantic similarity	Similarity of meaning of entities
2	*Descriptive* semantic similarity	Similarity of meaning of descriptions of entities (attributes and characteristics)
3	*Descriptive* semantic similarity of music albums	Similarity of meaning of descriptions of music albums
4	*Descriptive* semantic similarity of music albums in a music ontology (MusicBrainz)	Similarity of meaning of description of *MusicBrainz:album*(s) i.e. similarity of artist, title, label etc.
5	Semantic similarity of release year of albums from a music ontology	Similarity of meaning of numbers representing years (e.g. numerical similarity)
6	Semantic similarity of release year of albums from MusicBrainz ontology	Similarity of meaning of *MusicBrainz:Release_event:date*

choice of granularity should be made to best help solve a given problem, but very high granularities are not advised.

Implementation

Implementation of dimensions may vary greatly. For instance the *lexical* dimension may be implemented as a string edit distance like in the ASMOV [9] (that uses the Levenshtein distance), or as an *external* thesaurus lookup, like in ASCO [102] (which actually uses both the edit distance and WordNet similarity). As explained before, low granularity leaves a lot of room for different implementations.

In case of ontology matching *taxonomic* and *descriptive* dimensions are often combined into one, called *structural*. There are many different approaches to *structural* similarity. For instance, CIDER [100] uses a feature vector model that combines taxonomy and roles into one set of features. In Anchor-Flood [101], on the other hand, the *structural* similarity is constructed purely from taxonomy. ASMOV [9] has an even more disparate definition of *structural* dimension that involves a weighted sum of the domain and range similarities of roles. This difference of approaches demonstrates the importance of a good description of semantic similarity dimensions. Since *structural* similarity (dimension) lacks a good description, it allows for very different implementations. One possible definition, i.e. a dimension that combines *taxonomic* and *descriptive* similarities would endow it with a very low *granularity* that places it very close to a vague "universal" semantic similarity. In other words the meaning of *structural* similarity is too vague (it is very different

in each of the presented examples) and, therefore, it does not provide much information.

Let us reiterate that similarity dimensions are defined primarily by their interpretation and not by implementation, or even type of method used.

1.9 Applications of dimensional semantic similarity

In this paper, we have focused on presenting the idea of similarity dimensions on the examples concerning pairwise comparison of ontological entities. The idea itself can be applied to comparison of other objects, such as full ontologies, entities in semantic graphs, documents, etc. Throughout the text we have already suggested potential applications outside of ontological entities. Let us now reiterate and summarize these considerations.

Analysis of multiple articles and surveys on ontology matching [14, 13, 16] reveals that modern methods usually use multiple kinds of semantic similarity akin to similarity dimensions. In particular there is a strong distinction between *lexical* methods (also called *linguistic*) and others. A popular approach, exemplified in FalconAO [104] is to use *lexical* similarity first, as an input for further parts of the matching algorithm that use some kind of *structural* data (a graph matching algorithm in case of FalconAO). Some methods, such as AgreementMaker [105] and COMA [106] use multiple so-called matchers, some of which use taxonomy, relationship graph or lexical data. Matchers that work in the same dimension use different algorithms (e.g. some lexical matchers use edit distance, some thesaurus lookup or others). It seems that researchers in the field of ontology matching realized that construction of a good matching requires one to look at similarity of ontologies from many different perspectives. We have formalized this idea in the form of semantic similarity dimensions.

In the field of document analysis, semantic similarity means the similarity of meaning (in natural language) of the content of the documents i.e. text similarity [17]. Within this field, similarity of other features of documents, such as author, type of document (e.g. scientific article, a poem, news article, short story, etc.), publishing events and others is usually not considered. Those features are a good candidate for implementation of similarity dimensions (e.g. type of document describes the *taxonomic* dimension), but require *external* ontology (e.g. a taxonomy of document types), so, in some way, similarity of documents is understood as *lexical* similarity of content of documents.

The *lexical* dimension, in the context of document similarity, has many features that may be used to construct subdimensions. Features considered in practice [19] include statistical analysis (e.g. bag of words approach), sentence length, punctuation count, specific names count, synonyms, hypernyms, hyponyms and others. Those may be divided into corpus-level (e.g. TF-IDF), document-level (e.g. bag of words), sentence-level (e.g. extraction of subjects and objects, number of capitalized words) and word-level (e.g. synonyms, edit distance). Phrase-level features are also

sometimes considered, although they are used in machine translation [18] rather than in similarity scoring. Even though many researchers have proposed multiple features [19], so far there was no attempt to group those features into classes that would resemble the low granularity dimensional approach described in this paper. Although those are not applied to similarity scoring, there are many dimension-like properties relevant to text and speech analysis. Those include affect [20] (also applied to WordNet [21]), salience, writing style (formal or informal) and others. Theoretically, we might score text with respect to, for instance, affect similarity (dimension), but as said before, such high level properties are not included in current document similarity scoring methods.

Presence of similarity dimensions in semantic graphs is most pronounced in methods that use WordNet. This semantic graph offers many different kinds of edges and sets of features. Different methods use different subsets of available information (e.g. some methods use synsets, others bots synsets and homonyms). This dimensionality is, however, not made explicit and, so far, those methods have not been categorized with respect to dimensions.

1.10 Concluding remarks

The notion of (semantic) similarity is, by its nature, vague and ambiguous. Many semantic similarity measuring methods have been proposed and work well for ontology-specific or domain-specific applications. Their approaches, however, do not easily generalize across domains (or ontologies). The proposal of *similarity dimensions* address this problem and attempts at rectify the ambiguity of similarity scores.

A single, universal, score suggests how similar two entities are, but does not answer the question: in what way are the entities actually similar? A similarity vector provides such answers by treating each similarity dimension separately. Thus, it is possible to capture the fact that being *descriptively* similar is different from *taxonomicaly* similar, or *lexically* similar, etc. In short, similarity dimensions add extra meaning to similarity. Dimensional scores specify not only how similar entities are, but also why.

Generally speaking, there are two ways of dealing with semantic similarity. First, the overall approach, based on application of similarity dimensions, with separate scores in each, to understand how similar entities are, and in what way. Second, development of domain/ontology specific methods that focus on the nature of the problem at hand. The latter approaches (e.g. [63]) work well when solving a specific problems, but do not transfer well to other application areas. Canonically, similarity calculating methods produce a single score that combines all aspects of semantic similarity. It is a useful simplification that enables direct comparison of results from different methods. However, different methods approach similarity from a "different perspective," use different data and capture different aspect of semantic similarity.

Moreover, since any well-defined method is formally correct, no individual score can be said to be formally wrong.

Note also that, comparison of single number results form different methods is, by nature, flawed. Even methods that utilize multiple intermediate similarity scores, in the end provide a single weighted sum, which "flattens" the meaning of similarity. Furthermore, making explicit the considered aspect of semantic similarity can be also useful. For instance, Resnik's method is purely taxonomic. Hence, by explicitly labeling it as such, one gains valuable information. For instance, someone not familiar with details of Resnik's method would not know why similarity does not change, even if one adds a number of roles into the KB. Labeling the method as taxonomic informs that it is insensitive to roles.

On the other hand, the proposed dimensional similarity vector presents a more detailed (expanded) view of similarity, and allows for a more meaningful comparison of results between methods. Another advantage of similarity dimensions is that each one of them has a universal interpretation, that may be refined depending on context and is independent of the data format. As long as this interpretation is preserved, multiple different algorithms may be used to represent each dimension. Furthermore, proper usage and interpretation of a specific dimension is reliant on intuitive understanding of general description of that dimension. In this way, the similarity vector reflects the subjective nature of similarity.

Let us recall that there are many different and correct ways to model any given domain or problem. The multiplicity of modeling paradigms is a well-known and studied subject [59, 60]. It suggests that, for any domain, there is no single, exclusively correct, modeling solution. We believe that the same is true for semantic similarity, i.e. the correct "absolute" / "ultimate" similarity measure does not exist. Instead, the similarity changes with the perspective, from which we calculate it. Here, it should be stressed that the proposed approach recognizes this fact by its inherent flexibility. Specifically, it allows: (i) existence of domain/ontology specific methods to combine separate scores into a single one (as in [6]), (ii) restricting similarity dimensions that are actually considered in a given domain (e.g. only taxonomic and compositional dimensions are to be used), based on the "nature of the application." Moreover, if one is interested in similarity in a taxonomy, one needs to use only a taxonomic method. Alternatively, if one already obtained a dimensional vector, (s)he can utilize any part of it that is of interest in a given context. Here, again, available dimensionality provides information useful both before and after similarity scoring.

In this way, the dimensional similarity vector provides, in a sense, a disentanglement of similarity. A dimensional answer to a question of similarity is more informative not just because one receives more values, but also because each value (i.e. each dimension) has an interpretation. This interpretation adds knowledge about the way, in which entities are similar, on top of a numerical value representing similarity. A single score is much more concise, but it lacks this additional information, i.e. this information is obfuscated, when only one score is available, without any explanation as to how it was arrived at.

In summary, similarity dimensions are a way of introducing *semantics* the into semantic similarity itself. The low-granularity dimensions (presented in section 1.5) provide a basic understanding of similarity even to a layman. For instance *physical* similarity is immediately understood by everyone. High-granularity dimensions may be created to serve very particular needs of experts in a given field. It is thus our opinion that, in calculating semantic similarity, the most important part is the reason why we calculate it. In conclusion, recognizing similarity dimensions adds meaning to semantic similarity. Dimensional score tells us not only how entities are similar, but also indicates why.

References

1. Szmeja, P., Ganzha, M., Paprzycki, M., & Pawlowski, W. (2016, February). Dimensions of ontological similarity. In Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on (pp. 246-249). IEEE.
2. Jaccard, P. (1901) Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. Bulletin de la Société Vaudoise des Sciences Naturelles 37, 241-272.
3. Zhang, Lin, et al. "FSIM: a feature similarity index for image quality assessment." Image Processing, IEEE Transactions on 20.8 (2011): 2378-2386.
4. https://www.w3.org/TR/owl-guide/
5. E. Albacete, J. Calle, E. Castro and D. Cuadra, "Semantic Similarity Measures Applied to an Ontology for Human-Like Interaction", *Journal Of Artificial Intelligence Research*, Volume 44, p. 397-421, 2012
6. F. J. Calle, E. Castro, D. Cuadra, "Ontological Dimensions Applied to Natural Interaction" *ONTORACT '08 Proceedings of the 2008 First International Workshop on Ontologies in Interactive Systems*, p. 91-96
7. C. H. Papadimitriou and K. Steiglitz. "Combinatorial Optimization : Algorithms and Complexity". Prentice-Hall, 1998
8. OntologyAlignmentEvaluationInitiativehttp://oaei.ontologymatching.org/
9. "Ontology Matching with Semantic Verification" Yves R. Jean-Mary 1, E. Patrick Shironoshita 1, and Mansur R. Kabuka 1,2 1 INFOTECH Soft, Inc., 9200 S Dadeland Blvd. Suite 620, Miami, FL 33156, USA 1 University of Miami, Coral Gables, FL 33124, USA
10. M. Vargas-Vera, M. Nagy, E. Motta, "DSSim - managing uncertainty on the semantic web", 2011, pp. 1–11 URL (http://oro.open.ac.uk/23598/1/10.1.1.104.99635B15D.pdf).
11. R. Ichise, Machine learning approach for ontology mapping using multiple concept similarity measures, in: Seventh IEEE/ACIS International Conference on Computer and Information Science (icis 2008), Portland/Oregon, IEEE.
12. Euzenat, Jérôme, and Petko Valtchev. "Similarity-based ontology alignment in OWL-lite." ECAI. Vol. 16. 2004.
13. J. Euzenat, P. Shvaiko, Ontology Matching, Springer-Verlag, Berlin, 2007.
14. Shvaiko, Pavel, and Jérôme Euzenat. "Ontology matching: state of the art and future challenges." Knowledge and Data Engineering, IEEE Transactions on 25.1 (2013): 158-176.
15. Jeff Heflin, Dezhao Song. 2016. Ontology Instance Linking: Towards Interlinked Knowledge Graphs. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16).
16. Otero-Cerdeira, Lorena, Francisco J. Rodríguez-Martínez, and Alma Gómez-Rodríguez. "Ontology matching: A literature review." Expert Systems with Applications 42.2 (2015): 949-971.

17. Mihalcea, Rada, Courtney Corley, and Carlo Strapparava. "Corpus-based and knowledge-based measures of text semantic similarity." AAAI. Vol. 6. 2006.
18. Li, Junhui, Philip Resnik, and Hal Daumé III. "Modeling Syntactic and Semantic Structures in Hierarchical Phrase-based Translation." HLT-NAACL. 2013.
19. Murphy, M. Lynne. Semantic relations and the lexicon: antonymy, synonymy and other paradigms. Cambridge University Press, 2003.
20. Besnier, Niko. "Language and affect." Annual review of anthropology 19 (1990): 419-451.
21. Strapparava, Carlo, and Alessandro Valitutti. "WordNet Affect: an Affective Extension of WordNet." LREC. Vol. 4. 2004.
22. Erhard Rahm and Philip Bernstein. "A survey of approaches to auto-matic schema matching." VLDB Journal, 10(4):334–350, 2001
23. Younes, Amine Ait, Frederic Blanchard, and Michel Herbin. "New similarity index based on the aggregation of membership functions through OWA operator." Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on. IEEE, 2015.
24. Yager, R., On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE Trans. Systems, Man and Cybern., 18(1), pp. 183–190 (1988)
25. Bohm, C., Berchtold, S. and Keim, D.A., Searching in high-dimensional spaces: index structures for improving the performance of multi-media databases, ACM Computing Surveys, 33(3), pp. 322–373 (2001)
26. Semantic Similarity in Biomedical Ontologies Catia Pesquita, Daniel Faria, Andre O. Falca, Phillip Lord, Francisco M. Couto
27. Navigli, Roberto. "Word sense disambiguation: A survey." ACM Computing Surveys (CSUR) 41.2 (2009): 10.
28. Bunescu, Razvan C., and Marius Pasca. "Using Encyclopedic Knowledge for Named entity Disambiguation." EACL. Vol. 6. 2006.
29. Lund, Kevin, and Curt Burgess. "Producing high-dimensional semantic spaces from lexical co-occurrence." Behavior Research Methods, Instruments, and Computers 28.2 (1996): 203-208.
30. D. Bollegala, Y. Matsuo, M. Ishizuka. "A Relational Model of Semantic Similarity between Words using Automatically Extracted Lexical Pattern Clusters from the Web", *Conference on Empirical Methods in Natural Language Processing*, EMNLP 2009, p. 803-812. ACL and AFNLP, 2009
31. S. Harispe, D. Sánchez, S. Ranweza, S. Janaqia, J. Montmaina, "A framework for unifying ontology-based semantic similarity measures: A study in the biomedical domain", *Journal of Biomedical Informatics*, Vol. 48, p. 38-53, 2014
32. B. Suntisrivaraporn, "A Similarity Measure for the Description Logic EL with Unfoldable Terminologies", *2013 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, p. 408-413, 2014
33. Wan, S., & Angryk, R. A. (2007). Measuring Semantic Similarity Using WordNet-based context vectors. In M. El-Hawary (Ed.), IEEE International Conference on Systems, Man and Cybernetics, SMC 2007 (pp. 908–913). Montreal, Quebec, Canada: IEEE Computer Society
34. S. Harispe, S. Ranwez, S. Janaqi, J. Montmain, "Semantic Measures for the Comparison of Units of Language, Concepts or Instances from Text and Knowledge Representation Analysis", *CoRR*, 2013
35. R. Rada, H. Mili, E. Bicknell, M. Blettner, "Development and application of a metric on semantic nets" *Systems, Man and Cybernetics, IEEE Transactions on*, vol.19, no.1 , p. 17-19, 1989
36. Aha, D., Kibler , D., and Albert, M. (1991). Instance-Based Learning Algorithms. Machine Learning,6(1):37–66.
37. Z. Wu, M. Palmer, "Verb semantics and lexical selection" *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, p. 133-138, 1994
38. D.S. Blough, "Discrimination of letters and random dot patterns by pigeons and humans", *Journal of Experimental Psychology: Animal Behavior Processes*, Vol 11(2), p. 261-280, 1985

39. S.K. Rhee, J. Lee, M-W. Park, M. Szymczak, G. Frąckowiak, M. Ganzha, M. Paprzycki "Measuring Semantic Closeness of Ontologically Demarcated Resources", *Fundamenta Informaticae*, Vol.96, no.4, p. 395-418, 2009
40. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider, "The Description Logic Handbook" *Cambridge University Press*, 2003
41. W. Cohen, A. Borgida, H. Hirsh, "Computing Least Common Subsumers in Description Logics", *Proceedings of the 10th national conference on artificial intelligence*, MIT Press, p. 754-760, 1992
42. F. Baader, "Least Common Subsumers and Most Specific Concepts in a Description Logic with Existential Restrictions and Terminological Cycles", 2003
43. G. A. Miller, W. G. Charles, "Contextual correlates of semantic similarity" *Language and Cognitive Processes*, vol.6 p. 1-28. 1991
44. D. Milne and I. Witten, "An effective, low-cost measure of semantic relatedness obtained from wikipedia links," in Proc. of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, 2008, pp. 25–30.
45. A. Hliaoutakis, G. Varelas, E. Voutsakis, E. G. Petrakis, and E. Milios, "Information retrieval by semantic similarity," IJSWIS, vol. 2, no. 3, pp. 55–73, 2006.
46. D. Sanchez, M. Batet, D. Isern, and A. Valls, "Ontology-based semantic similarity: A new feature-based approach," Expert Systems with Applications, vol. 39, no. 9, pp. 7718–7728, 2012.
47. D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani, "Learning relatedness measures for entity linking," in Proceedings of the 22nd ACM international Conference on Information & Knowledge Management, 2013, pp. 139–148
48. A. Tversky, "Features of Similarity" *Psychological Review*, vol.84, p. 327-352, 1977
49. D. Lin, "An Information-Theoretic Definition of Similarity", *In Proceedings of the Fifteenth International Conference on Machine Learning*, p. 296-304, 1998
50. P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy", *Proceedings of the 14th international joint conference on Artificial intelligence*, p. 448-453, 1995
51. Pirró, G. (2009). A semantic similarity metric combining features and intrinsic information content. Data & Knowledge Engineering, 68, 1289–1308.
52. Pirró, G., & Seco, N. (2008). Design, Implementation and Evaluation of a New Semantic Similarity Metric Combining Features and Intrinsic Information Content. In R. Meersman & Z. Tari (Eds.), OTM 2008 Confederated International Conferences CoopIS, DOA, GADA, IS, and ODBASE 2008 (Vol. 5332, pp. 1271–1288). Monterrey, Mexico: Springer Berlin/Heidelberg.
53. Herbert Rubenstein and John Goodenough. Contextual cor-relates of synonymy. CACM, 8(10):627–633, October 1965.
54. Shannon, Claude Elwood. "A mathematical theory of communication." ACM SIGMOBILE Mobile Computing and Communications Review 5.1 (2001): 3-5
55. J. J. Jiang, D. W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy". *International Conference on Research on Computational Linguistics*, 1997
56. C. d'Amato, N. Fanizzi, F. Esposito, "A Semantic Similarity Measure for Expressive Description Logics", *Proceedings of convegno italiano di logica computazionale*, 2005
57. C. Chang, R. Lee, "Symbolic Logic and Mechanical Theorem Proving", Academic Press, San Diego, 1973
58. N. Fanizzi, C. d'Amato, F. Esposito, "Structural Similarity in Expressive Description Logics An Extended Family of Kernels for OWL", 2013
59. Goldkuhl, Göran. "Design theories in information systems-a need for multi-grounding." JITTA: Journal of Information Technology Theory and Application 6.2 (2004): 59.
60. Dietz, Jan LG. "What is Enterprise Ontology?". Springer Berlin Heidelberg, 2006.
61. C. Fellbaum, "WordNet: An Electronic Lexical Database", MIT Press, Cambridge, UK 1998
62. M. Ashburner,C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, et. al., "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." *Nature genetics* vol. 25, no. 1., p. 25-29. Department of Genetics, Stanford University School of Medicine, California, USA, 2000

63. S. Benabderrahmane, M. Smail-Tabbone, O. Poch, A. Napoli, M-D. Devignes, "IntelliGO a new vector-based semantic similarity measure including annotation origin", *BMC Bioinformatics*, vol.11, no.1, 2010
64. V. Pekar V, S. Staab, "Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision" *Proceedings of 19th International Conference on Computational Linguistics*, p. 1-7, 2012
65. I. Horrocks, U. Sattler, "A description logic with transitive and inverse roles and role hierarchies", *Journal of Logic and Computation*, vol.9 no.3, p. 385-410, 1999
66. Google Knowledge Graph`https://developers.google.com/structured-data/customize/overview`
67. Open Directory Project `https://www.dmoz.org/`
68. `http://linkeddata.org/`
69. `http://purl.oclc.org/NET/ssnx/ssn`
70. `https://en.wikipedia.org/wiki/Short-beaked_common_dolphin`
71. `https://en.wikipedia.org/wiki/Silvertip_shark`
72. `http://www.flmnh.ufl.edu/fish/gallery/descript/silvertipshark/silvertipshark.html`
73. `http://www.marinebiodiversity.ca/skatesandrays/external%20anatomy/ampullae%20of%20lorenzini.htm`
74. `https://en.wikipedia.org/wiki/Lesser_electric_ray`
75. `http://www.arkive.org/lesser-electric-ray/narcine-brasiliensis/`
76. Cunningham, P., A Taxonomy of Similarity Mechanisms for Case-Based Reasoning, IEEE Trans. Knowledge and Data Engineering, Vol. 21 (11), pp. 1532–1543, (2009)
77. Zerzucha, P. and Walczak, B., Concept of (dis)similarity in data analysis, Trends in Analytical Chemistry, 38, pp. 116–128 (2012)
78. Detyniecki, M., Mathematical aggregation operators and their application to video querying, Research Report, LIP6, Paris (2001)
79. Dubois, D. and Prade, H., On the use of aggregation operations in information fusion processes, Fuzzy Sets and Systems, 142, pp. 143–161 (2004)
80. K. Lehmann, "A Framework for Semantic Invariant Similarity Measures for ELH Concept Descriptions", Diplomarbeit, Technishe Universitat Dresden, 2012
81. A.G. Maguitman, F. Menczer, H. Roinestad, A. Vespignani, "Algorithmic Detection of Semantic Similarity", *Proceedings of the 14th International Conference on World Wide Web* p. 107-116, 2005
82. T. Bach, R. Dieng-Kuntz, Measuring Similarity of Elements in OWL DL ontologies, Workshop on Contexts and Ontologies: Theory, Practice and Applications, 2005.
83. M.A. Rodriguez, M.J. Egenhofer, "Determining semantic similarity among entity classes from different ontologies", *IEEE Transactions on Knowledge and Data Engineering 15*, p. 442-456, 2003
84. E.G.M. Petrakis, G. Varelas, A. Hliaoutakis, P. Raftopoulou, "X-Similarity:Computing Semantic Similarity between Concepts from Different Ontologies", *Journal of Digital Information Management*, vol.4, p. 233-237, 2006
85. Semantic Measures Library `http://www.semantic-measures-library.org/sml/`
86. Shepard, Roger N. (1962). "The analysis of proximities: Multidimensional scaling with an unknown distance function. I.". Psychometrika. 27 (2): 125–140. doi:10.1007/BF02289630.
87. Hahn, Ulrike; Chater, Nick; Richardson, Lucy B (2003). "Similarity as transformation". Cognition. 87 (1): 1–32. doi:10.1016/S0010-0277(02)00184-1.
88. Asl, Mohamad Eydani, et al. "Similitude Analysis of Composite I-Beams with Application to Subcomponent Testing of Wind Turbine Blades." Experimental and Applied Mechanics, Volume 4. Springer International Publishing, 2016. 115-126.
89. Nothdurft, Hans-Christoph. "Feature analysis and the role of similarity in preattentive vision." Attention, Perception, & Psychophysics 52.4 (1992): 355-375.

90. Santini, Simone, and Ramesh Jain. "The graphical specification of similarity queries." Journal of Visual Languages & Computing 7.4 (1996): 403-421.
91. De Nies, Tom, et al. "A distance-based approach for semantic dissimilarity in knowledge graphs." Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on. IEEE, 2016.
92. http://davis.wpi.edu/xmdv/datasets/ohsumed.html
93. http://qwone.com/~jason/20Newsgroups/
94. Hamming, Richard W. (1950), "Error detecting and error correcting codes", Bell System Technical Journal, 29 (2): 147–160, doi:10.1002/j.1538-7305.1950.tb00463.x, MR 0035935.
95. Zhou, Zili, Yanna Wang, and Junzhong Gu. "A new model of information content for semantic similarity in WordNet." Future Generation Communication and Networking Symposia, 2008. FGCNS'08. Second International Conference on. Vol. 3. IEEE, 2008.
96. Seco, Nuno, Tony Veale, and Jer Hayes. "An intrinsic information content metric for semantic similarity in WordNet." Proceedings of the 16th European conference on artificial intelligence. IOS Press, 2004.
97. Sánchez, David, Montserrat Batet, and David Isern. "Ontology-based information content computation." Knowledge-Based Systems 24.2 (2011): 297-303.
98. Blank, A. "Words and concepts in time: Towards diachronic cognitive onomasiology". In R. Eckardt, K. von Heusinger & C. Schwarze (Eds.), Words in time (pp. 37–66). Berlin, Germany: Mouton de Gruyter, 2013.
99. MusicBrainz - The Open Music Encyclopedia https://musicbrainz.org/
100. Jorge Gracia and Kartik Asooja, "Monolingual and Cross-lingual Ontology Matching with CIDER-CL: Evaluation Report for OAEI 2013", Proc. of 8th Ontology Matching Workshop (OM'13), at 12th International Semantic Web Conference (ISWC'13), Syndey (Australia), CEUR-WS, ISSN-1613-0073, volume 1111, October 2013.
101. Seddiqui, Md Hanif, and Masaki Aono. "Anchor-flood: results for OAEI 2009." Proceedings of the 4th International Conference on Ontology Matching-Volume 551. CEUR-WS. org, 2009.
102. Le Bach Thanh, Dieng-Kuntz Rose, and Gandon Fabien. Ontology matching: A machine learning approach for building a corporate semantic web in a multi-communities organization, April 14-17 2004.
103. Feiyu, Lin. State of the art: automatic ontology matching. Tekniska Högskolan, 2007.
104. Hu, Wei, and Yuzhong Qu. "Falcon-AO: A practical ontology matching system." Web Semantics: Science, Services and Agents on the World Wide Web 6.3 (2008): 237-239.
105. Cruz, Isabel F., Flavio Palandri Antonelli, and Cosmin Stroe. "AgreementMaker: efficient matching for large real-world schemas and ontologies." Proceedings of the VLDB Endowment 2.2 (2009): 1586-1589.
106. Massmann, Sabine, et al. "Evolution of the COMA match system." Proceedings of the 6th International Conference on Ontology Matching-Volume 814. CEUR-WS. org, 2011.