Department of
Information Engineering
and Computer Science
DISI

UNIVERSITY
OF TRENTO - Italy

DISI - Via Sommarive 14 - 38123 Povo - Trento (Italy)
http://www.disi.unitn.it

Entity Type Recognition – dealing with the Diversity of Knowledge

FaustoGiunchiglia, MattiaFumagalli

September 2020

Technical Report # DISI- 20-008

Published in: Conference on Knowledge Representation and Reasoning, 2020

Entity Type Recognition – dealing with the Diversity of Knowledge

Fausto Giunchiglia , Mattia Fumagalli

Department of Information Engineering and Computer Science (DISI) - University of Trento, Italy

{fausto.giunchiglia, mattia.fumagalli}@unitn.it

Abstract

Semantic Heterogeneity is the problem which arises when multiple resources present differences in how they represent the same real world phenomenon. In KR, an early approach was the development of ontologies and, later on, when ontologies showed at the knowledge level the same semantic heterogeneity that they were meant to fix at the data level, to compute mappings among them. In this paper we acknowledge the impossibility of avoiding semantic heterogeneity, this being a consequence of the more general phenomenon of the *diversity* of the world and of the world descriptions. In this perspective the heterogeneity of ontologies is a *feature* (and not a bug to be fixed by aligning them) which gives the possibility to use the *most suitable* ontology in any given application *context*. The main contributions of this paper are: (i) a novel articulation of the problem of semantic heterogeneity, as it appears at the knowledge level, as *contextuality*, (ii) its qualitative and quantitative formalisation in terms of a set of *diversity* and *unity metrics* and (iii) an *Entity Type Recognition* algorithm which selects the contextually most appropriate ontology and exploits it to solve the current problem, e.g., the alignment and integration of a set of input schemas. The experimental results show the validity of the approach.

1 Introduction

Semantic Heterogeneity is the problem which arises when multiple resources, e.g., knowledge or data bases, usually developed by independent parties, present differences in the way they represent the same real world phenomenon. In KR, the first and main approach was and, correctly so, still is, the development of a set of reference *Knowledge Bases (KBs)*, e.g., ontologies or schemas, to be used as reference resources.[1] This line of work has produced a high quantity of high quality results. As an example, LOV, LOV4IoT, and DATAHUB,[2] three among the most relevant repositories, collectively contain around 1000 such KBs, some of which contain thousands of elements. In turn, this work has motivated the research on *Ontology* and *Schema Alignment* (Euzenat, Shvaiko, and others 2007; Giunchiglia and Shvaiko 2003; Giunchiglia, Yatskevich, and Shvaiko 2007; Shvaiko and Euzenat 2011; Algergawy et al. 2018)[3] with the goal of absorbing the differences across reference KBs.

In this work we exploit and build upon the work mentioned above, but taking a somewhat alternative approach, as originally envisaged in (Giunchiglia 2006). The starting point is the observation that the heterogeneity of data and knowledge is a special case of the more general phenomenon of *diversity*. Diversity is a distinguishing feature of the *world*: there will never be two identical moments places or objects. This generates *incompleteness* and this is the main motivation for data or knowledge integration in its various forms, see, e.g., (Giunchiglia and Fumagalli 2019; Wang et al. 2017). At the same time, diversity is also pervasive in the *descriptions* of the world: even for the same phenomenon, different observers will provide different descriptions, in relation to the local *context*, i.e, needs, objectives, and many other factors (Giunchiglia and Fumagalli 2017), this being the main bottleneck towards integration.[4]

The diversity of descriptions can be split in two largely independent phenomena. The first is the *diversity of language*, namely the fact that the mapping between words and their intended meaning is many-to-many (Giunchiglia, Batsuren, and Bella 2017), as witnesses by many well-known linguistics phenomena, e.g., *polisemy*, *homographs*, *synonymity*. The second is the *diversity of knowledge*, namely the fact, even under the assumption of no language diversity, that there is a many-to-many mapping between entity types (*etypes* from now on) and the properties used to describe them. Thus, we may have etypes which are under-specified, etypes which are over-specified and cumulate the properties of less general etypes, as in Schema.org[5] (Patel-Schneider 2014) and KBs where the same etypes have largely disjoint sets of properties, as it is the case with Schema.org and DBpedia.[6]

In this paper we propose a solution to the problem of

[1] In this paper we use the terminology used for knowledge graphs and represent KBs as sets of *schemas* and, in turn, *schemas* as *types of entities*, e.g., *Person*), each being associated with a set of *properties* (e.g., *birth-date*) (Bonatti et al. 2019; Kejriwal 2019). In other words, a KB is a knowledge graph with no instances and where the nodes are entity types and the links are properties.

[2] https://lov.linkeddata.es/, http://lov4iot.appspot.com/, https://old.datahub.io/

[3] See also http://om2019.ontologymatching.org/#ap

[4] Notice how semantic heterogeneity, as defined above, is a special case of the diversity of the world descriptions.

[5] www.schema.org

[6] wiki.dbpedia.org

the diversity of knowledge.[7] Technically we instantiate this problem as an *etype recognition* problem as follows:

Given multiple reference KBs and given an input KB with a set of unknown etypes, each associated to a set of known properties, predict all the etypes of the input KB.

Some observations. The *first* is that, as from above, the diversity of knowledge appears in the form of two or more heterogeneous etypes which are indeed different descriptions of possibly the same real world phenomenon. The *second* is that we assume the availability of a repertoire of high quality (heterogeneous) reference KBs, e.g., those from the repositories mentioned above. Given the unavoidability of diversity, the heterogeneity of the reference KBs is a *feature* (and not a bug to be fixed by aligning them) in that different reference KBs encode different application *contexts*. The *third* is that diversity becomes a problem *only if and when it is unknown and needs to be locally handled*, for instance when there is a new KB (e.g., of a set of schemas which need to be aligned). The *fourth* is that the solution is not to eliminate the diversity of the input KB, which is impossible, but, rather, to understand how the new KB relates to the existing reference KBs. The *fifth* is the assumption that the etypes of the input KB are unknown, even if the input KB defines them. This technically models the unknown quality and diversity of the input KB.

The algorithm proposed in this paper handles the heterogeneity of the input KB in two steps: (i) select the KB which codifies the closest contextual view point and (ii) use it to disambiguate the etypes of the input KB based on their properties. Notice how *any manipulation (e.g., composition) of the reference KBs, if at all needed, is done only based on the knowledge of the input KB.* The main contributions of this paper are as follows:

1. a *formal model* of knowledge diversity, where knowledge is modeled as a set of semantically heterogeneous KBs, where KBs are modeled as a (variation of) *Formal Concept Analysis (FCA) contexts* (Ganter and Wille 2012) (Section 2);
2. a *graphical model* and representation of *contextuality*, in terms of *Knowledge Lotuses* (Section 2);
3. the articulation of contextuality in terms of the *unity* and *diversity*, where unity is defined in terms of what is shared across contexts, etypes or properties, and the opposite for diversity (Section 3);
4. a *quantitative model* of knowledge diversity and unity, as a *set of metrics* which apply to contexts, etypes and properties (Section 4);
5. an *Entity Type Recognition* algorithm, implemented as a classifier applied to the reference KB which better fits the input schema (Section 5).

The paper is completed as follows. Section 6 provides an evaluation of the algorithm presented in Section 5. Section 7 and Section 8 provide the related work and the conclusions, respectively.

[7] In KBs, language diversity appears in the labels used to denote entity types and properties. Section 5 describes how we handle language diversity using techniques from (Giunchiglia, Yatskevich, and Shvaiko 2007).

2 Diversity as contextuality

We adapt ideas and notation from FCA (Ganter and Wille 2012) and formalize KBs, which we take to be sets of etypes and corresponding properties, as *(Knowledge) Contexts*, where we define a (Knowledge) Context C as $C = \langle E_C, P_C, I_C \rangle$, with $E_C = \{e_1, ..., e_n\}$ being the set of *etypes*, $P_C = \{p_1, ..., p_n\}$ being the set of *properties* of C, and I_C being $I_C = \{\langle e, I_C(e) \rangle \mid e \text{ is an etype of } C\}$, with $I_C(e) = \{p \in P_C \mid p \text{ is a property of } e\}$. We say that an etype is *associated to* a property when the latter is used to describe the former, and that a property is *associated to* an etype with the dual meaning.[8] These definitions are similar to the ones from FCA with two key differences:

- E_C is a set of etypes and *not* a set of entities;
- I_C applies to a single element of E_C and *not* to a subset of E_C.

Table 1: *A context for (a portion of) SUMO*

SUMO (representation) context		Properties						
		service	*date*	*part*	*value*	*birthdate*	*citizen*	*cohabitant*
Etype	sumo:Organization	×						
	sumo:FinancialAccount		×					
	sumo:GeopoliticalArea	×	×					
	sumo:AbstractEntity			×				
	sumo:CollateralEntity				×			
	sumo:ServiceProcess	×						
	sumo:Object			×				
	sumo:Person					×	×	×

Thus, for instance, Table 1 reports a set of etypes (left) with corresponding properties (top) from SUMO.rdf [9]., Version 1.0. The value boxes with crosses represent I_C, while the set of value boxes in the same line represent $I_C(e)$ for the etype e in that same line. A missing cross means that that property is not used to describe that etype. The main motivation for this choice is that etypes are the basic elements which populate a knowledge context. Here, the word *"populate"* is used on purpose, meaning that etypes have for contexts the same role that entities have for etypes. In the same way as (the schema of) a single etype collects entities at the data level, (the schema of) a single knowledge context collects etypes at the knowledge level; and, in both cases, properties are what allows to discriminate among elements, i.e., etypes or entities. From a philosophical point of view this intuition is rooted in the recent work in *Teleosemantics* (Macdonald, Papineau, and others 2006; Millikan 2017), as largely discussed in (Giunchiglia and Fumagalli 2016), which shows that there is no real difference between etypes (i.e., classes), called *Kinds* in (Giunchiglia and Fumagalli 2016), and etypes (i.e., instances), called *Individuals* in (Giunchiglia and Fumagalli 2016).

Based on the above intuitions, we call E_C the *Extent of the knowledge context C* and P_C the *Intent of the knowledge context C*. Furthermore, we define the notion of *(Knowledge)*

[8] Notice how *etypes* are defined in terms of properties which are taken to be primitive.A straightforward extension would be to reason about properties defined in terms of "more primitive" properties. This is part of the future work.

[9] www.adampease.org

Concept Co as the pair $Co = \langle e, I_C(e) \rangle$. When $p \in I_C(e)$ we also talk of e being in the domain of p, in formulas $e \in dom(p)$. $I_C(e)$ is called the *Intent of the Concept Co* as it defines how its etype is *intentionally* defined in terms of a subset of the properties of C. Notice that $I_C = \{Co_i\}_{i \in M}$, with M the number of etypes of C. In other words, a context is a set, actually a lattice, of concepts (Ganter and Wille 2012).

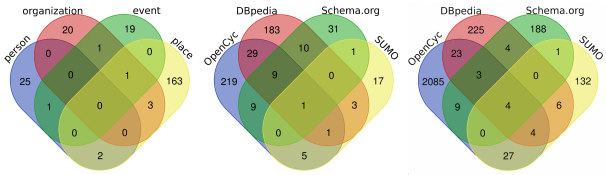
Given our reference scenario, we make two assumptions:

- We have an unbound number of contexts $C_i = \langle E_{C_i}, P_{C_i}, I_{C_i} \rangle$ which, in turn, ...
- ... can make use of any number of etypes E_{C_i} and properties P_{C_i}, some of which are never used in any other context.

These hypotheses model what we call a *Diverse KB*, characterized by the impossibility of making any design time assumption about the number and (diversity of) content of the available contexts. We model this situation in that we allow a possibly *infinite* set of etypes E, a possibly *infinite* set of properties P, and a set of N contexts C_i, with $\bigcup_i E_{C_i} \subseteq E$ and $\bigcup_i P_{C_i} \subseteq P$, where K, defined as:

$$K = \bigcup_{i \in N} C_i,$$

is the system's KB at any given time. Following the terminology from (Giunchiglia 1993) we say that K is a *Multi-Context system*.



(a) SUMO (b) Person (c) All properties

Figure 1: *(a) Shared properties across etypes; (b) Shared properties across resources; (c) Shared etypes across resources.*

Given K, we represent the diversity which occurs within and across its constituent contexts with *Knowledge Lotuses*. Knowledge lotuses are *Venn Diagrams*.[10] Fig 1 depicts three lotuses where we assume that we have four contexts built from (parts of) the four biggest KBs from the repositories mentioned above, namely OpenCyc (OC)[11], the DBpedia (schema) (DB), Schema.org (SH), and SUMO (SU). These three lotuses are paradigmatic examples of *all and only* the possible visualizations of knowledge diversity. In fact, knowledge lotuses allow for the *hierarchical modeling* of the three core elements of knowledge, (*viz.*, contexts, etypes, and properties), namely: *(a)* contexts, for each context, *(b)* its etypes and, for each etype, *(c)* its properties. The key intuition is to fix one of these three elements (namely the context(s), or the etype(s), or the property(ies)) and then to study the diversity of the second against the third. Each combination provides a different perspective on diversity. Let us analyse Fig 1.

- Lotus (a) fixes the context (SUMO) and it represents the diversity of etypes in terms of their (un)shared properties. The dual case of comparing properties in terms of the etypes in their domain is also possible. These types of lotuses represent the *diversity internal to a context, in terms of their etypes or their properties.*
- Lotus (b) fixes the etype (*Person*) and it represents the diversity of contexts in terms of their (un)shared properties (for that etype). The dual case of comparing properties in terms of the contexts where they occur is also possible. These types of lotuses represent the *diversity across contexts, for any given etype*.
- Lotus (c) fixes the properties (considering all of them) and it represents the diversity of contexts in terms of the (un)shared etypes. The dual case of comparing etypes in terms of the contexts where they occur is also possible. These types of lotuses represent the *diversity across contexts, for any given property*.

Thus, for instance, looking at Fig 1(a), in SUMO, *Person* and *Place* share 2 property terms, while they are distinguished by 26 and 167 terms, respectively. Looking at Fig 1(b), the four representations of *Person* share only one property, while two of them, i.e., OpenCyc and DBpedia share 40 properties.

3 Context, unity and diversity

We model KBs as contexts, where each context encodes a different *viewpoint* on the world. The issue is how to model the diversity which occurs *across* and *within* contexts, in the latter case, across their etypes and properties. Let us assume that we want to study the diversity of any two homogeneous knowledge elements, *viz.* two contexts, two etypes, or two properties. Then, given any two such elements, e.g., two etypes,

- Their *(Etype) Diversity* relates
  1. to how many properties they *do not share* within one or more (or all) contexts and
  2. to how many contexts they *do not share*, in relation to one or more (or all) of their properties.
- Their *(Etype) Unity*[12] relates
  1. to how many properties they *do share* within one or more (or all) contexts and
  2. to how many contexts they *do share*, in relation to one or more (or all) of their properties

Some observations. The *first* is that the definitions above can be generalized to contexts and properties. The *second* is that the etype diversity and unity labeled with [1.] correspond to the Lotus (a) in the previous section, while those

[10] Lotuses in Fig 1 represent four sets. Simpler/complex lotuses can be depicted to represent the diversity of lower/ higher numbers of resources. More complex lotuses (more than 6 intersections) can be visualized using *ad-hoc* data visualization libraries (e.g., the upset-module from https://asntech.shinyapps.io/).

[11] www.cyc.com

[12] The notion of Unity used here is unrelated to the notion, with the same name, used in OntoClean (Guarino and Welty 2002).

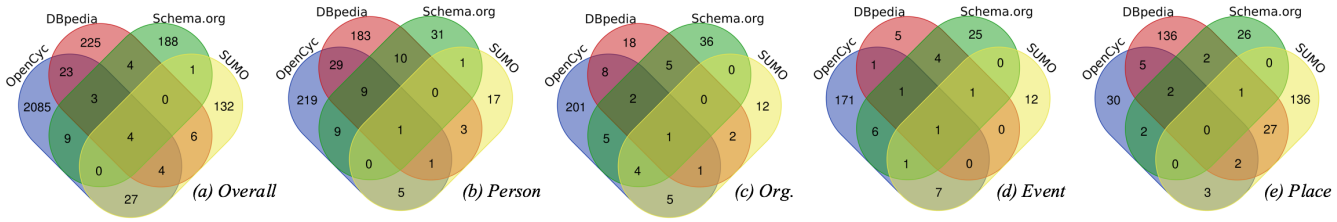


Figure 2: *(a) etypes across contexts; (b) (c) (d) (e) properties of Person, Organization, Event and Place across contexts.*

labeled with [2.] are the dual case of Lotus (c). In fact the above definitions of *diversity* and *unity* are *all and only* the possible cases and can be mapped one-to-one with the lotuses from the Section 2. The *third* is that *diversity* and *unity* are not binary properties but they take a *continuum* of values. The *fourth* is that Diversity and Unity are not opposite properties. For instance, two etypes can have both a high level of diversity (because of a lot of unshared properties or contexts) and a high level of unity (because of a lot unshared properties or contexts). Any diversity measure D and its complement unity measure U can be assumed to stand in the relation $D+U=1$ only if we define a reference scenario, for instance, if we know K with its set of entities and properties. But, as from Section 2, we assume that this information is not available (e.g., because of the evolution in time of contexts).

Fig. 2 provides a first comparison, where Fig. 2(a) provides the total number of (un)shared etypes across the four resources, while Fig. 2(b), 2(c), 2(d), 2(e) provide the number of (un)shared properties of *Person*, *Organization*, *Event* and *Place*. Furthermore, Table 2 reports the etypes shared across, 2, 3, 4 contexts, while Table 3 reports the properties shared across the four contexts for *Person*.

The first observation is about the *low level of unity* across contexts. Despite the fact that they are all supposed to be general purpose there are only four etypes which are shared by all of them: *Event*, *Place*, *Person* and *Organization*, namely the etypes for time and space, i.e., the two a priori of perception (Kant 1998; Strawson 2017) and the arguably most common types of *agenthood*. Notice that, if we add the fifth and sixth biggest contexts, i.e., Proton[13] and YAGO[14], we still maintain the same four shared etypes. Things change if we consider smaller contexts, the main reason being that these latter contexts are focused on specific aspects of the world; for instance, FOAF[15] is focused on *Person*.

The second observation is about the *high level of diversity* across contexts. As from Fig. 2(a) most etypes are defined in only one context (e.g., ~ 2000 in OpenCyc and ~ 220 in DBpedia) this being mainly motivated by the different focus. Thus, for instance, Schema.org is more focused on information objects while DBpedia contains a large amount of information about biological species, with this phenomenon becoming even more evident in smaller contexts (e.g., FOAF).

The third observation is about the *diversity and unity of properties*. Consider for instance the properties of *Person*

[13]http://www.ontotext.com/proton/protontop.html

[14]https://datahub.io/collections/yago

[15]http://www.foaf-project.org/

Table 2: *The shared etypes across knowledge contexts*

contexts	***Tot.***	***etypes***
OC, DB, SH, SU	4	*Person, Organization, Event, Place, . . .*
OC, DB, SH	3	*SportsTeam, SportsEvent, Action, . . .*
OC, DB, SU	4	*City, PoliticalParty, Language, Animal, . . .*
OC, DB	23	*Hospital, SportsLeague, BodyOfWater, Sport, . . .*
OC, SH	9	*Offer, EducationalOrganization, Message, Role, . . .*
OC, SU	27	*UnitOfMeasure, CreditAccount, ComputerNetwork, . . .*
DB, SH	4	*VideoGame, CreativeWork, Airport, . . .*
DB, SU	6	*Region, Ship, MilitaryUnit, Agent, . . .*
SU, SH	1	*Vehicle, . . .*

Table 3: *The shared properties of Person*

contexts	***Tot.***	***Properties***
OC, DB, SH, SU	1	*spouse*
OC, DB, SH	9	*date, title, number, related, birth, parent, work, name, place*
OC, DB, SU	1	*occupation*
OC, DB	29	*ethnicity, skin, activity, employer, status, education*
OC, SH	9	*contact, suffix, tax, job, children, works, worth, gender, net*
DB, SH	10	*death, sibling, point, member, nationality, award, parents*

as from Table 3. On one side, the *most shared* properties are those which seem, somewhat often, relevant while, on the other side, the *least shared* are those which seem less relevant. Thus, for instance, the most shared properties of *Person* are, e.g., *name*, *birth*, *place*, *occupation*, or *title*, while, for instance, in Schema.org (only), *Person* has a huge amount of properties concerning their business, e.g., *sponsor*, *brand* or *catalog*. Similarly, in DBpedia (only), *Person* has more biologically relevant properties, e.g., *blood*, *body* or *race*. The first type of properties are somewhat related to *essential* or *rigid* properties, as defined in (Guarino and Welty 2002). Furthermore, following Donellan (Donnellan 1966), these properties are those which are more amenable for an *attributive usage* while the others are more amenable for a *referencial* (or *contextual*) *usage*. Notice, however, that this distinction is not clear-cut. No property is fully rigid in the sense that there will always be contexts which are outliers. For instance a person living alone in the jungle will have no name or, following the example in (Guarino and Welty 2002), a dead person may have had her brain removed. *All properties are contextual, still with different degrees of rigidness/referentiality*, and this applies also for those occurring in reference knowledge contexts that we decide to take as a priori knowledge. We capture this idea by calling the two types of properties mentioned above, standing on somewhat opposite extremes, *(quasi) rigid* and *(highly) contextual*, respectively.

4 Knowledge metrics

But how to use *diversity* and *unity* to select the most suitable context? By metrics.

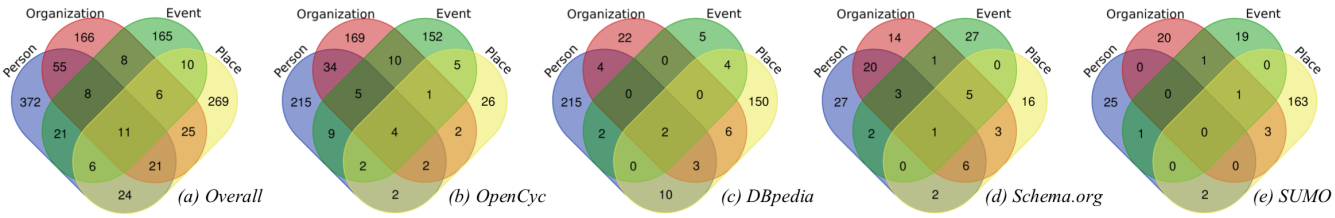


Figure 3: *(a) properties shared among four etypes across contexts; (b) (c) (d) (e) properties shared among four etypes in each context.*

4.1 Etype and property metrics

We start from property metrics. Towards this goal, we rely on the seminal work by Eleanor Rosch and, in particular, on her notion of *cue validity* (Rosch 1999; Rosch and Mervis 1975). This notion was defined as "*the conditional probability* $p(c_j|f_j)$ *that an object falls in a category* c_j *given a feature, or cue,* f_j". It was used to define the set of basic level categories, namely those categories which maximize the number of characteristics shared by their members (what here we call highly contextual properties) and minimize the number of characteristics shared with the members of their sibling categories (what here we call quasi-rigid properties) (Rosch 1999). The intuition is that *basic level categories* have higher cue validity and, because of this, they are easier to recognize. Rosch's definitions were designed for experiments where humans (trained to recognize objects because of their life experiences) were asked to identify objects based on their visual properties. In our setting a pre-trained classifier is asked to recognize an etype based on its properties. We follow Rosch's original methodology and define the *cue validity of a property p w.r.t to an etype e*, also called $cue_p - validity$, as

$$Cue_p(p,e) = \frac{PoE(p,e)}{|dom(p)|} = c \in [0,1] \quad (1)$$

with $|\ X\ |$ being the cardinality of the set X and *PoE(p, e)* being defined as:

$$PoE(p,e) = \begin{cases} 1, if\, e \in dom(p) \\ 0, otherwise \end{cases} \quad (2)$$

$Cue_p(p,e)$ returns 0 if p is not associated with e and *1/n*, where n is the number of etypes in the domain of p, otherwise. In particular, if p is associated to only one etype its $cue_p - validity$ is maximum and equal to one. The intuition is that all properties have the same recognition potential (which we assume to be normalized to one) and that this impact is equally "divided" across the etypes they are associated to: the more etypes, the more widespread the impact and the lower the impact per etype. $Cue_p(p,e)$ is a *diversity metric* and, as such, it grows whenever properties are not shared across entities. Given the notion of $cue_p - validity$ we define the notion of cue validity of an etype, also called $cue_e - validity$, as the sum of the cue validities of the properties associated with the etype, namely:

$$Cue_e(e) = \sum_{i=1}^{|prop(e)|} Cue_p(p_i,e) = c \in [0, prop(e)] \quad (3)$$

Table 4: *Cue values computed from the lotuses in Figure 3, with a focus on Person and Event.*

	Person			**Event**		
	Cue_e	Cue_{er}	Cue_{ec}	Cue_e	Cue_{er}	Cue_{ec}
(a) Overall	**436.3**	**0.84**	**0.16**	**193.85**	**0.82**	**0.18**
(b) OpenCyc	**241.47**	**0.88**	**0.12**	**167.64**	**0.89**	**0.11**
(c) DBpedia	**224.49**	**0.95**	**0.05**	**10.5**	**0.81**	**0.19**
(d) Schema.org	**42.22**	**0.69**	**0.31**	**31.39**	**0.80**	**0.20**
(e) SUMO	**26.5**	**0.95**	**0.05**	**20.33**	**0.92**	**0.18**

where *prop(e)* is the set of properties which are associated with e. The intuition is the same as Rosch's: the etypes with higher $cue_e - validity$ will be the easiest to recognize.

However the $cue_e - validity$ does not tell us anything about the *level of contextuality* or, dually, of *rigidness* of an etype, meaning by this how many properties it shares with other etypes in the same context, a parameter with major implications on knowledge recognition. To make an example, assume that we have two etypes and two properties, with two possible situations: (i) both etypes share both properties and (ii) the two etypes are each associated to one property. In both cases the $cue_e - validity$ of the two etypes is one but, while in the first case the two etypes are indistinguishable, in the second case they are highly identifiable. In other words, having a high $cue_e - validity$ the two etypes are highly recognizable but, having a low level of rigidness, they are hardly distinguishable. We capture this intuition via the notion of *(level of) rigidness* of an etype, also called $cue_{er} - validity$, as:

$$Cue_{er}(e) = \frac{Cue_e(e)}{|prop(e)|} = c \in [0,1] \quad (4)$$

Thus, for instance, in the first case in the example above, the $cue_{er} - validity$ of the both etypes will be 0.5 while in the second case it will be one. Dually, we define *the (level of) contextuality of an etype*, also called $cue_{ec} - validity$, as

$$Cue_{ec}(e) = 1 - Cue_{er}(e) = c \in [0,1] \quad (5)$$

The cue_{er}-validity is a diversity metric while the cue_{ec}-validity is a *unity metric* in that it grows with the number of shared properties. To understand why we have called the above metrics, diversity of unity metrics, Look at Table 4, which reports the cue values extracted from the knowledge lotuses represented in Figure 3, with a focus on *Person* and *Event*. A first observation is that the value of Cue_e tells us that provides insights about the size of its etype. For instance, excluding the overall view, the context with the largest etypes is OpenCyc, while the value of Cue_{er} is connected to the number of properties in the petals of the lo-

tuses in Figure 3 (e.g., 372 in Person (a) and 215 in Person (b)). Looking at the data in Table 4 It can be noticed how Schema.org is the weakest in terms of *rigidness* and the best in level of *contextuality* (see the Cue_{ec} value). Finally, see how the accumulation of properties of a given entity type through multiple contexts results in a decrease of the Cue_{er} value and an increase of the Cue_{ec} value.

In graphical terms, the relation between lotuses and the cue metrics can be understood as follows: the higher the number of the properties in the petals of a lotus, the higher the value of Cue_{er} (which is in fact a diversity metric) and, dually, the higher the number of the properties in the corolla of a lotus, the higher the values of Cue_{ec} (which is in fact a unity metric). This is the same when lotuses are used to represent the unity and diversity in terms of shared (and not shared) etypes. Thus, lots of elements in the corolla means unity, while in the petal mean diversity.

4.2 Context metrics

The notions and terminology used for etypes, i.e., the notions of Cue_e and Cue_{er} and Cue_{ec} can be generalized to contexts, generating context diversity and unity metrics, as follows:

$$Cue_c(C) = \sum_{i=1}^{|E_C|} Cue_e(e_i) = |prop(C)| \quad (6)$$

$$Cue_{cr}(C) = |prop(C)| / \sum_{i=1}^{|E_C|} prop(e_i) = c \in [0,1] \quad (7)$$

$$Cue_{cc}(C) = 1 - Cue_{cr}(C) = c \in [0,1] \quad (8)$$

The result reported in equation (6) can be easily understood by looking at equations (1), (3): etypes only get distributed the recognition potential of properties which is anyhow fixed by their number. While the *cue validity of a context* (i.e., its number of properties), also called $cue_c - validity\ Cue_c(C)$, will measure the overall capability of a context to support knowledge recognition, the *rigidness validity of a context*, also called $cue_{cr} - validity\ Cue_{cr}(C)$, derived by equation (7), will measure its overall level of rigidness. These two cues model the properties of a context. However, no matter its quality, a context is useless if it does not represent the properties of the etype to be predicted. Let us formalize this intuition. Let I be an input context, which we assume to be a set of sets of properties, where each set describes an unknown etype. Then, we define the *coverage of a context C*, also called *ci-coverage*, w.r.t to an input schema I as follows:

$$Cov_{ci}(C,I) = 1 - \left(\frac{|prop(I) - prop(C)|}{|prop(I)|} \right) = c \in [0,1] \quad (9)$$

where the function *prop* is extended to apply to both C and I. If $Cov_{ci}(C,I) = 1(0)$, then all (no) properties in I occur in C. Notice how coverage is a unity metric. Furthermore, we take into account the fact that the non-shared properties play no role in the knowledge recognition task, for any such task the actual knowledge recognition capability will be highly dependent on the input. We take this into account by introducing the two notions of *(input) relative cue validity of a context*, in formulas $Cue_{ci}(C,I)$, and *(input) relative rigidness of a context*, in formulas $Cue_{cri}(C,I)$[16] as follows:

$$Cue_{ci}(C,I) = \sum_{i=1}^{|E_{C\cap I}|} Cue_e(e_i) = |prop(C \cap I)| \quad (10)$$

$$Cue_{cri}(C,I) = |prop(C \cap I)| / \sum_{i=1}^{|E_{C\cap I}|} prop(e_i) = c \in [0,1] \quad (11)$$

These two definitions are obtained from equations (6) - (7) by substituting C with $C \cap I$, where $C \cap I$ is obtained from C by deleting all the properties not shared with I, and also all those etypes which, as a result, have zero properties. We call $C \cap I$ the *shared context*. Notice how the two metrics above are, respectively, diversity and unity metrics of the shared context.

5 The etype recognition algorithm

The introduction has informally introduced the main idea underlying the implementation of the etype recognition algorithm. Based on that, the algorithm is implemented in two steps, as follows:

1. *Context determination:* select the best reference context and check its suitability and, in case such a context has been found,
2. *eType recognition in context:* predict the unknown etypes.

Let us consider these two steps in detail. The first step, namely selection of the best context and the validation of its appropriateness for the given task is performed based on the set of metrics described in Section 4. This component implements two main steps:

1. *maximize the unity* between the input context and the reference context. We implement this requirement by selecting those contexts where the coverage is highest;
2. *maximize the diversity* of the reference context. We implement this requirement by selecting the context generating the context, shared with the input, with the highest relative cue validity.

The underlying intuition should be obvious: with the first operation we minimize the misalignment between the reference context and the input context, thus minimizing the amount of noise introduced by the enrichment, while, with the second operation, we minimize the noise introduced by the context internal confusion across etypes.

Let us now concentrate on the second step, which we implement as a *multi-label supervised classification* task. The process can be briefly described as follows:

1. All reference contexts are expressed in the *Terse RDF Triple Language (Turtle)*[17] format. The context hierarchy is flattened into a set of sets of triples, where each triple encodes information about "etype-property" associations $I_C(e)$ (e.g., the triple "Person-domainOf-friend"

[16] $Cue_{ei}(C,I)$ and $Cue_{eri}(C,I)$ are used also to relativise the notions of $cue_e - validity$ and $cue_{er} - validity$. Technically this can be done by taking etypes to be single etype contexts.

[17] https://www.w3.org/TR/turtle/

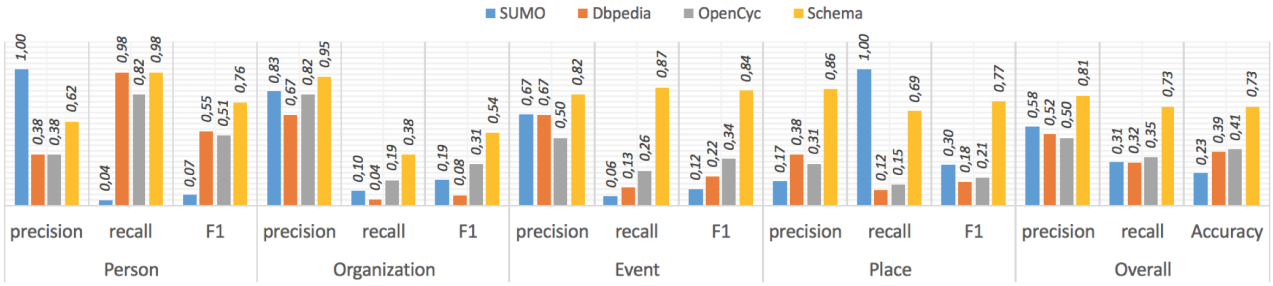


Figure 4: *Precision/recall/F1, for Person, Org., Event, Place in SUMO, DBpedia, OpenCyc and Schema.org, + Overall Accuracy.*

encodes the "Person-friend" $I_C(e)$ association. Notice that this representation maps to the FCA encoding described in Section 2 (see, e.g., Table 1), and, according to it, each relation of type "Person-friendOf-Person" is expressed by two triples, i.e., "Person-domainOf-firendOf" and "Person-rangeOf-firendOf". Moreover, this method resembles the underlying idea of *Knowledge Graph Embedding* (KGE) methods (Wang et al. 2017; Dumančić, García-Durán, and Niepert 2019), which consists of finding the vector representation of triples like "Marc-friendOf-Eve". The main difference in KGE is that the vectors provide representation of instances (and values). Differently, in our approach we provide a representation of sort of *"higher-order predications"*, where the relations (i.e., the predicates of the triples) always denote meta-level relations (e.g., domainOf) between the graphs concepts. This encoding is applied to all the selected reference contexts as well as to the input contexts.

2. The context labels (both properties and etypes) present a high level of syntactic and semantic heterogeneity, that is, many such labels are minor variations of the same label, still carrying the same meaning. We handle this problem exploiting techniques which are very similar to those used in ontology matching, see, e.g., (Giunchiglia and Shvaiko 2003; Giunchiglia, Autayeu, and Pane 2012; Bella, Giunchiglia, and McNeill 2017). The key idea is that labels are analyzed via a NLP pipeline which performs various steps, including, for instance: a) split a string every time a capital letter is encountered (e.g., *birthDate* $\rightarrow$ birth and date); b) lower case all characters; c) filter out stop-words (e.g., *hasAuthor* $\rightarrow$ author), d) substitute synonyms.
3. Each selected reference context is used to train a model later used to predict the etype(s) of the input context. In the training, the etypes of the reference context are used as ground truth.

It is important to notice our quite unusual notion of ground truth. As a matter of fact we have one ground truth per reference context. From a technical point of view this is made possible by the very high quality of the reference contexts. There is in fact a sense in which each of them codifies a different local truth, i.e., that implicitly defined by the resource creators. This highlights the fact that the diversity among contexts is nothing else that the syntactic representation of different diverse local meanings and semantics (for instance as codified in multi-context systems (Ghidini and Giunchiglia 2001)). Once we take knowledge diversity for real, we have to give up the idea that we have a single (ground) truth to which we can report for the final decision of what is the case.

6 Evaluation

We organize the evaluation in three parts. In Section 6.1 we analyse the effects that the internal diversity and unity of contexts has on their ability to enable the recognition of their own etypes. This section is quite important as it shows the negative effects which (may) arise with the size of contexts. The next two sections evaluate the two components of the etype recognition algorithms. In particular, in Section 6.2 we analyze the performance of the etype recognition algorithm, while in Section 6.3 we analyse the performance of the context determination algorithm.

The experiments have been done using decision trees applied to the four contexts selected in Sect. 3. We selected accuracy as main criterion, with max depth possible. We pruned the tree after the generation, by replacing some branches according to confidence "0.25". Notice that, because of the high quality of the KBs data, learned models and parameters selection were very robust to changes, thus providing indirect evidence of the generality of the results. When training the models, we have used only properties, thus forgetting the objects of properties. Finally, in the evaluation, we have used *precision*, *recall*, *F1 measure* and *accuracy*, the latter used with even distributions of properties across etypes.[18]

6.1 Etype self-recognition

The question is the contexts' ability to (self)-recognize their own etypes. In this experiment we have used the four etypes present in all four contexts, namely, *Person*, *Organization*, *Event* and *Place*. The results are reported in Table 5.

Table 5: *Results of the first experiment*

Trial	*Accuracy*	*Person F1*	*Organization F1*	*Event F1*	*Place F1*
OpenCyc vs. OpenCyc	0.98	0.98	0.98	0.99	0.97
DBpedia vs. DBpedia	0.98	0.99	1.00	0.90	0.99
Schema.org vs. Schema.org	0.73	0.76	0.53	0.84	0.76
SUMO vs. SUMO	1.00	1.00	1.00	1.00	1.00

Differently from what we expected, F1 is almost never 1.0, with two negative peaks with Schema.org (on all etypes) and DBpedia (on *Event*). Furthermore, as expected, the more etypes are used in the training phase, the worse the performance gets. This phenomenon is well explained by how

[18] Data can be found at https://github.com/knowdive/ETR

diversity operates. Ideally, one would like to use the biggest possible context. But the main (only?) way to achieve this is to add contextual properties: the more diversity the more ability to discriminate. However, the side effect is an increase of the probability of sharing contextual properties across etypes which, in specific contexts, have a similar role. This effect is quite evident in Schema.org, where *Organization* and *Person* share properties related to their agentive roles. Dually all SUMO's scores are top, this being motivated by the fact that most SUMO properties are quasi-rigid.

6.2 Etype recognition in context

In this experiment we have used the same models as in the first experiment but with input etypes only from Schema.org. The results are reported in Fig. 4. For each trial (one per reference context), we provided *Accuracy*, *F1*, *Precision* and *Recall*, this allows us to better understand and exploit the role of each reference context in the etype recognition task. The highest scores are, of course, always with Schema.org. Let us focus on the behaviour of the other reference contexts. Let us focus on *Overall* (extreme right). In terms of *accuracy*, the second best reference context is OpenCyc. This means that with OpenCyc we have the highest proportion of *true results* among the total number of cases examined (e.g., among all the results of the ETR task). Notice that we have a *true result* when a property *is* (or *is not*) a property of an etype according to the input context and it is recognized as such, given the reference context. However the situation changes when we analyze *Precision* and *recall*. Considering *precision*, the second best reference context is SUMO, meaning by this that with SUMO we have the highest proportion of properties recognized as properties of given etypes, according to the reference context, that are *truly* properties of that etypes according to the input context (e.g., high precision for *Person* and *Organization* means that a high number of properties recognized as properties of *Person* and *Organization* are properties of *Person* and *Organization* according to the input context). while, considering *Recall*, OpenCyc is again the second best.

This misalignment between the different measures is quite relevant in this setting. In fact, here the situation is opposed with respect to what is usually the case in machine learning: instead of having a single ground truth against which we evaluate a set of case studies, here we have a single case study (in Fig. 4, there are five of them analyzed in parallel: four etypes and the overall combination) and a set of ground truths that are, actually, the goal of our evaluation. Depending on the goal we may prefer a *higher precision* (i.e., decreasing the false positives) in which case we maximize the unity, namely the coherence of the input context with the reference context. This is desirable in applications where the reference context is used as background knowledge, in data integration tasks, or when building the model in vertical federated learning tasks (Yang et al. 2019). Notice that, in these applications, an exceedingly *low precision* is to be avoided in that this means high confusion across the etypes of the two contexts. But in other applications, e.g., in horizontal federated learning tasks (Yang et al. 2019), we may prefer to have a *lower recall* (i.e., increasing the number of false negatives), in which case we maximize diversity, as this gives us more attributes over which to integrate examples. Notice that, in these applications, an exceedingly *high recall* is to be avoided as diversity would become too low and there would be no new properties to be considered. The choice of the context should be by the data scientist on the basis of her domain knowledge.

Let us now consider the single etypes in Fig. 4. Here the link between the results as from Fig. 4 and knowlege lotuses (see in particular Fig. 3) becomes quite explicit, thus suggesting the pivotal role of knowledge metrics (see the next subsection). Let us consider for instance the OpenCyc and SUMO *Organization* etypes. These etypes have, respectively, a huge and a small amount of contextual properties not shared with the other contexts (201 and 12). Both OpenCyc-Organization and Sumo-Organization share a small amount of quasi-rigid properties with the input Schema.org context, but the amount of shared properties in OpenCyc is more than double the amount of SUMO (12 vs. 5). Moreover, looking at Fig. 3(b) and (e) it is possible to observe that OpenCyc-Organization shares 58 out of 227 properties with the other etypes (i.e., about 25%) while SUMO-Organization shares 5 out of 25 properties with the other etypes (i.e., about 20%). This behavioural pattern occurs with the other etypes as well. We can therefore derive the following conclusions:

- the quasi-rigid properties of the reference context have a positive impact on precision;
- a large number of properties in the reference context has a positive effect on recall;
- the overlapping of properties across etypes in the reference context affects negatively both precision and recall, by increasing the number of false positives and false negatives;
- a high F1 score reflects both good precision and recall, and indicates a good trade-off between quasi-rigid properties, total amount of properties and low level of overlapping properties inside the reference context.

6.3 Context determination

The goal here is to show how the results discussed above can be correctly predicted by the metrics; and then, to analyze the role of the metrics in the context determination step.

Let us start with Sect. 6.1. The contexts in Table 5 are ranked exactly by their Cue_{cr} values in Table 7, where Schema.org has the lowest Cue_{cr}. Accordingly, the conclusion that can be drawn is that, under the assumption of full coverage (here each context is tested against itself), *the higher is the level of contextuality the lower is the level of confusion of contexts in their ability to recognize etypes*.

Let us now consider Sect. 6.2. From the overall *accuracy* results reported in Fig. 4 we can observe that the best reference context (Schema.org excluded) in the recognition of the *overall* set of Schema.org etypes is OpenCyc. This is mainly motivated by the fact that, as for the values reported in Table 6, OpenCyc is the reference context *that maximizes the unity with the input context* (see its Cov_{ci} score). Notice that OpenCyc is not the best context in the *maximiza-*

Table 6: *Metrics and F1 Score for SUMO, DBpedia, OpenCyc, DBpedia, Schema.org, Schema.org(+) and SUMO+DBpedia.*

	Context				***Person***				***Organization***				***Event***				***Place***			
	Cue_ci	*Cue_cri*	*Cov_ci*	***Acc.***	*Cue_ei*	*Cue_eri*	*Cov_ci*	***F1***	*Cue_ei*	*Cue_eri*	*Cov_ci*	***F1***	*Cue_ei*	*Cue_eri*	*Cov_ci*	***F1***	*Cue_ei*	*Cue_eri*	*Cov_ci*	***F1***
SUMO	20	0.86	0.15	**0.23**	2.5	0.83	0.02	**0.07**	6	0.85	0.06	**0.19**	3.50	0,87	0.03	**0.12**	8.00	0.88	0.07	**0.30**
Dbpedia	39	0.68	0.30	**0.39**	19.33	0.77	0.20	**0.55**	6.83	0.68	0.08	**0.08**	4.00	0.50	0.06	**0.22**	8.83	0.63	0.11	**0.18**
OpenCyc	52	0.74	0.38	**0.41**	22.91	0.76	0.24	**0.51**	13.41	0.70	0.15	**0.31**	9.75	0.81	0.09	**0.34**	5.91	0.65	0.07	**0.21**
Schema	127	0.68	1.00	**0.73**	42.22	0.69	0.48	**0.76**	30.87	0.58	0.42	**0.54**	31.39	0.80	0.31	**0.84**	22.38	0.68	0.26	**0.77**

Table 7: *Relativized and non-relativized context metrics*

	Cue_c	Cue_{ci}	Cue_{cr}	Cue_{cri}
SUMO	235	20	0.96	0.86
DBpedia	427	39	0.92	0,68
OpenCyc	638	52	0.87	0.74
Schema	127	127	0.68	0.68

tion of diversity, but approaches the best (namely, SUMO, see its Cue_{cri}). This fact highlights a stronger impact by Cov_{ci} with respect to Cue_{cri}, providing further evidence of what already suggested by the data in Fig. 3, as discussed in Sect. 6.2. The high accuracy of OpenCyc can also be further explained by checking the *Cov* scores for each etype in Table 6. The only etype for which OpenCyc is not the best is *Place*, this being explained by the low Cue_{eri} with respect to SUMO (which has the same Cov_{ci}). The suitability of each context can be also checked at the level of etypes. Looking at each *F1* result reported in Fig. 4, we can notice that the best performing etypes are those where the unity with the input context is maximized. However, even when the input is largely covered, a low level of diversity may lead to a decrease in performance, as in the case of Person-OpenCyc vs. Person-DBpedia and Place-OpenCyc vs. Place-SUMO (see Table 6).

The overall conclusion is that *coverage Cov_{ci} is the metric which has the highest impact while the second most important is Cue_{cri} (and Cue_{eri})*. This indicates that the algorithm for context determination implements the right strategy and also the possibility to improve its performance via the evaluaton of the best trade-off between the maximization of the unity and the maximization of diversity.

7 Related work

This work builds upon the large amount of work done in KR whose goal is the development of methodologies, tools, and actual ontologies aimed at the solution of the semantic heterogeneity problem, see, e.g., (Guarino and Welty 2002). The difference is in the approach. Following in spirit the Teleosemantics approach (Macdonald, Papineau, and others 2006), we believe that a general solution to the problem of knowledge diversity can only be provided in terms of a general *process* which adapts and evolves existing KBs based on the needs which appear in each and any single task, e.g., of data or knowledge integration. The work on *Teleologies* and *iTelos* (Giunchiglia and Fumagalli 2017; Giunchiglia and Fumagalli 2019), together with the work presented here, are first steps in this direction.

The work most similar in spirit is that on ontology matching, which has been largely cited in the introduction. There are however a few important differences. The most important is that in our case there is a reference KB which is taken as the ground truth. The alignment task is therefore not symmetric. This is the key intuition that allows us to exploit machine learning and, therefore all the huge amount of work developed in the area of knowledge embeddings (Wang et al. 2017). The results reported in Section 6.1, which could not be found in a symmetric approach, provide evidence of the advantages of the proposed approach.

Work on the problem of entity type recognition is described in (Sleeman and Finin 2013; Sleeman, Finin, and Joshi 2015). But, even if the problem is the same, motivation and approach are completely different. In fact, this work focuses on how to identify coreferent instances in heterogeneous semantic graphs where the underlying schemas are too general and not informative enough, and possibly even not known. The solution is based on the idea of exploiting the information codified in the instances populating the knowledge graph.

The idea of modeling knowledge as a set of contexts was independently proposed in (Giunchiglia 1993; Giunchiglia and Serafini 1994) and in (McCarthy 1993). In particular the idea of context proposed here is very similar to that described in (Giunchiglia 1993), where a context is taken to be that *"subset of the complete state of an individual that is used for reasoning about a given goal"*. Work on multicontext systems, is also described in (Brewka et al. 2018) where it is applied to reactive systems. But if the intuition underlying the notion of context is the same, the technical development in this earlier work is very different as it is the overall goal of the research.

The notion of *cue validity* has been widely studied, together with other similar measures such as "mutual information" and "category utility" with the general goal of measuring the informativeness of a category (Peng, Long, and Ding 2005), where these measures are pivotal in feature engineering (Witten et al. 2016). The key difference, which applies also to Rosch's original definitions, is that our notions apply at the knowledge level, on schemas, rather than on data. In this perspective, the notion of coverage has no equivalent in the data level metrics. This is a direct consequence of the fact that, differently from data, knowledge is designed to have general (re-)applicability and this is exactly what cue and coverage have been designed for.

8 Conclusion

In this paper we have proposed a formal model and a visual representation of knowledge, when it consists of a set of heterogeneous KBs. This in turn has allowed us to implement entity type recognition as a multi-label classification problem applied to a set of reference KBs. The future work will concentrate on a general methodology for re-using, adapting and evolving the large number of existing high-quality KBs.

9 Acknowledgments

This paper was supported by the *WeNet* project, funded by the European Union (EU) Horizon 2020 programme under grant number 823783.

References

Algergawy, A.; Cheatham, M.; Faria, D.; Ferrara, A.; Fundulaki, I.; Harrow, I.; Hertling, S.; Jiménez-Ruiz, E.; Karam, N.; Khiat, A.; et al. 2018. Results of the ontology alignment evaluation initiative 2018. In *13th International Workshop on Ontology Matching co-located with the 17th ISWC (OM 2018)*, volume 2288, 76–116.

Bella, G.; Giunchiglia, F.; and McNeill, F. 2017. Language and domain aware lightweight ontology matching. *Journal of Web Semantics* 43:1–17.

Bonatti, P. A.; Decker, S.; Polleres, A.; and Presutti, V. 2019. Knowledge graphs: New directions for knowledge representation on the semantic web (dagstuhl seminar 18371). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Brewka, G.; Ellmauthaler, S.; Gonçalves, R.; Knorr, M.; Leite, J.; and Pührer, J. 2018. Reactive multi-context systems: Heterogeneous reasoning in dynamic environments. *Artificial Intelligence* 256:68–104.

Donnellan, K. S. 1966. Reference and definite descriptions. *The philosophical review* 75(3):281–304.

Dumančić, S.; García-Durán, A.; and Niepert, M. 2019. A comparative study of distributional and symbolic paradigms for relational learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 6088–6094. AAAI Press.

Euzenat, J.; Shvaiko, P.; et al. 2007. *Ontology matching*, volume 18. Springer.

Ganter, B., and Wille, R. 2012. *Formal concept analysis: mathematical foundations*. Springer.

Ghidini, C., and Giunchiglia, F. 2001. Local models semantics, or contextual reasoning= locality+ compatibility. *Artificial intelligence* 127(2):221–259.

Giunchiglia, F., and Fumagalli, M. 2016. Concepts as (recognition) abilities. In *FOIS*, 153–166.

Giunchiglia, F., and Fumagalli, M. 2017. Teleologies: Objects, actions and functions. In *ER 2017*, 520–534. Springer.

Giunchiglia, F., and Fumagalli, M. 2019. On knowledge diversity. *Proceedings of the 2019 Joint Ontology Workshops, WOMoCoE*.

Giunchiglia, F., and Serafini, L. 1994. Multilanguage hierarchical logics, or: how we can do without modal logics. *Artificial intelligence* 65(1):29–70.

Giunchiglia, F., and Shvaiko, P. 2003. Semantic matching. *The Knowledge Engineering Review Journal* 265–280.

Giunchiglia, F.; Autayeu, A.; and Pane, J. 2012. S-match: an open source framework for matching lightweight ontologies. *Semantic Web* 3(3):307–317.

Giunchiglia, F.; Batsuren, K.; and Bella, G. 2017. Understanding and exploiting language diversity. In *IJCAI*, 4009–4017.

Giunchiglia, F.; Yatskevich, M.; and Shvaiko, P. 2007. Semantic matching: Algorithms and implementation. In *Journal on data semantics IX*. Springer. 1–38.

Giunchiglia, F. 1993. Contextual reasoning. *Epistemologia, special issue on I Linguaggi e le Macchine* 16:345–364.

Giunchiglia, F. 2006. Managing diversity in knowledge. 1. IOS Press. See also the site at http://knowdive.disi.unitn.it/ and the slides at http://www.disi.unitn.it/ fausto/knowdive.ppt.

Guarino, N., and Welty, C. 2002. Evaluating ontological decisions with ontoclean. *Communications of the ACM* 45(2):61–65.

Kant, I. 1998. Critique of pure reason (translated and edited by p. guyer & a. w. wood).

Kejriwal, M. 2019. *Domain-Specific Knowledge Graph Construction*. Springer.

Macdonald, G.; Papineau, D.; et al. 2006. *Teleosemantics*. Oxford University Press.

McCarthy, J. 1993. Notes on formalizing context.

Millikan, R. G. 2017. *Beyond concepts: Unicepts, language, and natural information*. Oxford University Press.

Patel-Schneider, P. F. 2014. Analyzing schema. org. In *International Semantic Web Conference*, 261–276. Springer.

Peng, H.; Long, F.; and Ding, C. 2005. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis & Machine Intelligence* (8):1226–1238.

Rosch, E., and Mervis, C. B. 1975. Family resemblances: Studies in the internal structure of categories. *Cognitive psychology* 7(4):573–605.

Rosch, E. 1999. Principles of categorization. *Concepts: core readings* 189.

Shvaiko, P., and Euzenat, J. 2011. Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering* 25(1):158–176.

Sleeman, J., and Finin, T. 2013. Type prediction for efficient coreference resolution in heterogeneous semantic graphs. In *2013 IEEE Seventh International Conference on Semantic Computing*, 78–85. IEEE.

Sleeman, J.; Finin, T.; and Joshi, A. 2015. Entity type recognition for heterogeneous semantic graphs. *AI Magazine* 36(1):75–86.

Strawson, P. F. 2017. *Subject and predicate in logic and grammar*. Routledge.

Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29(12):2724–2743.

Witten, I. H.; Frank, E.; Hall, M. A.; and Pal, C. J. 2016. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10(2):12.