A Framework for Semantic Similarity Measures to enhance Knowledge Graph Quality

Zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
der Fakultät für Wirtschaftswissenschaften
Karlsruher Institut für Technologie (KIT)

genehmigte
Dissertation
von

Dipl.-Ing. Ignacio Traverso Ribón

Tag der mündlichen Prüfung:	10. August 2017
Referent:	Prof. Dr. York Sure-Vetter
Korreferent:	Prof. Dr. Maria-Esther Vidal Serodio

Abstract

Precisely determining similarity values among real-world entities becomes a building block for data driven tasks, e.g., ranking, relation discovery or integration. Semantic Web and Linked Data initiatives have promoted the publication of large semi-structured datasets in form of knowledge graphs. Knowledge graphs encode semantics that describes resources in terms of several aspects or resource characteristics, e.g., neighbors, class hierarchies or attributes. Existing similarity measures take into account these aspects in isolation, which may prevent them from delivering accurate similarity values. In this thesis, the relevant resource characteristics to determine accurately similarity values are identified and considered in a cumulative way in a framework of four similarity measures. Additionally, the impact of considering these resource characteristics during the computation of similarity values is analyzed in three data-driven tasks for the enhancement of knowledge graph quality.

First, according to the identified resource characteristics, new similarity measures able to combine two or more of them are described. In total four similarity measures are presented in an evolutionary order. While the first three similarity measures, OnSim, IC-OnSim and GADES, combine the resource characteristics according to a human defined aggregation function, the last one, GARUM, makes use of a machine learning regression approach to determine the relevance of each resource characteristic during the computation of the similarity.

Second, the suitability of each measure for real-time applications is studied by means of a theoretical and an empirical comparison. The theoretical comparison consists on a study of the worst case computational complexity of each similarity measure. The empirical comparison is based on the execution times of the different similarity measures in two third-party benchmarks involving the comparison of semantically annotated entities.

Ultimately, the impact of the described similarity measures is shown in three data-driven tasks for the enhancement of knowledge graph quality: relation discovery, dataset integration and evolution analysis of annotation datasets. Empirical results show that relation discovery and dataset integration tasks obtain better results when considering semantics encoded in semantic similarity measures. Further, using semantic similarity measures in the evolution analysis tasks allows for defining new informative metrics able to give an overview of the evolution of the whole annotation set, instead of the individual annotations like state-of-the-art evolution analysis frameworks.

Acknowledgements

Even when only my name is written in the cover of this thesis, this document would not exist without the help of several people: Without listing everybody by name, I wish to thank these important persons in my life.
I would like to express my profound gratitude to **Prof. Dr. Rudi Studer** and **Prof. Dr. York Sure-Vetter** for allowing me to work in such a good environment and for their support and supervision during my PhD. I am also proud of having **Prof. Dr. Maria-Esther Vidal** as co-advisor. Thanks to her guidance and support I learn how to be more confident and trust into my own work.
I would like to name **Stefan Zander** and **Suad Sejdovic** as two of my best friends in Karlstuhe, who strongly supported me during my time at FZI and my beginning in Germany. Both of them are unconditional colleagues and friends, always available for helping with any kind of issue. Further, a deep thank you for **Heike Döhmer** and **Bernd Ziesche** for teaching me how does FZI works and how to speak and write German respectively.
Thanks also to all the people external to FZI and AIFB who helped me during my life and made possible that I am writing these lines today. Particularly, I would like to express my gratitude to **Laura Padilla**, who accompanied me during the last five years in the bad and good moments. Finally, I want to thanks my parents because thanks to their sacrifice, work, education and support from the distance I am here today.

Thank you very much / Muchas gracias a todos.

Karlsruhe, August 10, 2017 *Ignacio Traverso Ribón*

Contents

1 Introduction

Identity and *Difference* are terms whose meaning have been occupying the thoughts of philosophers throughout history. Aristotle formulated the Law of Identity in logic and this law is still valid today. The usual formulation is $A \equiv A$, which means that every thing is identical to itself. Two things are considered identical if they share completely the same properties. In contrary case, they are different. This binary relation is not enough for the understanding of the human being. Humans are able to estimate the degree of difference between two things in a fuzzy way. The contrary of the degree of difference is the degree of likeness, i.e., the similarity. Similarity is a fuzzy concept and depends on the properties shared among two things, the context and the personal background of the observer. The context implicitly weights the describing properties with a degree of relevance. For example, when comparing two job candidates, properties like studies or previous work experience are considered more relevant than the name or the address. The personal background of the estimator has also influence. An observer with superficial knowledge in music may think that the two candidates are similar because one of them loves classical music and the other one likes heavy metal. On a general level, both of them like music. However, an observer with deeper knowledge in music may see these two persons as totally different and even bind to them properties associated with these kinds of music like aggression or structured.

This work presents a novel semantic similarity measure framework composed of similarity measures able to consider several aspects or resource characteristics encoded in knowledge graphs simultaneously. The framework consists of four semantic similarity measures introduced in a evolutionary way according to the amount of considered resource characteristics or complexity. Additionally, the benefits of considering such semantics in data-driven tasks to enhance knowledge graph quality are shown. In total the framework is applied to three tasks namely 1. relation discovery among knowledge graph entities, where similarity measures are used to find discovery candidates, 2. knowledge graph integration, where semantic allows to better detect different representations of the same real world entity, and 3. evolution monitoring of annotation graphs, where semantics allows to define new informative metrics that give an overview of the evolution of the whole annotation graph instead of the individual annotations.

In the remainder of this chapter, the motivation of the work is explained. Three research questions give rise to three contributions presented in rest of the chapters. Previous publications supporting this work are enumerated and the structure of this document is described.

1.1 Motivation

In order to address the problem of determining relatedness between objects from a scientific perspective, scientists have defined several similarity measures for different purposes. Thus, the different properties or features of an object are translated into a mathematical representation, where the comparison is easily measurable. So, properties like dimensions, weight or even colors are today encoded in numbers representing different measurements such as meters, grams or proportion of additive primary colors (RGB coding). However, a measure is only a method to quantify the similarity. Humans were able to do comparisons before these measures were described. This is due to the conceptualization of the universe that is shared among members of a community. This conceptualization represents somehow a minimal agreement on the understanding of the world. Humans usually exchange their knowledge and therefore their understanding of the world through the language. A word stands for a concept within a community and every member in that community agree on the mapping between the word and the concept.

Computers do not own natively a conceptualization of the world and they cannot build it by themselves. Therefore, computers need to be provided with the conceptualization of a community, in order to be able to understand the different members. From the second half of the 20th century until today computer scientists have been providing such conceptualizations by means of ontologies. According to Studer et al. [96] an ontology is a formal, explicit specification of a shared conceptualization. Semantic Web and Linked Data initiatives have facilitated the definition of large ontologies and linked data datasets in form of knowledge graphs, e.g., Gene Ontology [28] or DBpedia [55]. Thus, computers have now a deeper understanding about the data they are processing and can use this understanding to estimate more accurate similarity values. Several tasks like search or data reuse have already benefited from the use of semantic technologies. However, the benefits have still not reached all the fields in computer science, for example, most of machine learning algorithms. Though similarity measures are widely used among machine learning algorithms, most of them omit knowledge encoded in ontologies or knowledge graphs, i.e., semantics. Traditionally, computers compare categorical properties in a binary way, i.e., 1 or 0. Semantics encoded in knowledge graphs and ontologies can alleviate this issue, enabling similarity measures to be aware of the meaning of such attributes and so improve the efficiency of machine learning algorithms.

Moreover, modern Web search engines make use of knowledge graphs to represent information extracted from the web in a structured way, e.g., the Google Knowledge Graph [93]. Knowledge graphs enable search engines to provide better results to search queries. However, knowledge graphs also suffer from classical data quality problems like incompleteness or isolation (lack of integration). Thus, quality problems in knowledge graphs may affect also the quality of the search results for a certain search engine. Semantic similarity measures can take into account the semantics encoded in knowledge graphs and so alleviate such quality issues.

This thesis identifies which aspects of similarity can be observed in knowledge graphs and which of them are already considered by existing semantic similarity measures. Furthermore, we define a framework containing four new similarity measures able to cover the identified aspects and evaluate their efficiency in three data-driven tasks related to the enhancement of knowledge graph quality.

1.2 Research Questions and Contributions

This section describes the three research questions answered by this thesis and the corresponding contributions.

The increasing amount and size of available ontologies and linked data datasets enable communities to enrich the descriptions of their data with semantics. For example, Gene Ontology terms are extensively used for capturing functional information of proteins and genes as showed in the UniProt-GOA database [38]. However, state-of-the-art similarity measures do not consider all the semantics available in such ontologies. We hypothesize that considering semantics encoded in ontologies and knowledge graphs can improve the accuracy of similarity values among knowledge graph resources. Thus, the following research question is formulated:

- **Research Question 1**: What is the improvement in terms of accuracy of considering semantics during the computation of similarity between two knowledge graph resources?

To address the first research question we describe three similarity measures that consider different types of information or aspects encoded in knowledge graphs and ontologies. The results generated by these similarity measures are compared with 11 state-of-the-art similarity measures included in a well-known benchmark.

Considering semantics encoded in knowledge graphs may increase the complexity of similarity measures. Given that similarity measures are usually computed in near real-time applications like search engines or recommendation systems, it is necessary to reach a trade-off among the cost in terms of time and the improvement in terms of accuracy when considering such complex information. This trade-off is addressed in the following research question:

- **Research Question 2**: How can semantic similarity measures efficiently scale up to large datasets and be computed in real-time applications?

For the purpose of addressing the second research question we study both the theoretical and the empirical complexity of the similarity measures. The results show that reducing the complexity of the considered semantics allow to decrease the computational complexity and keep high quality results.

Similarity measures may be used to alleviate knowledge graph quality issues as lack of integration or incompleteness. Isolated descriptions from the same entity in different knowledge graphs impede search engines to recognize that both descriptions represent the same real world entity. Thus, the search engines cannot offer a full view of the corresponding entity to the user. Similarly, knowledge graph may not be complete, being some relations between the graph entities omitted. Data-driven tasks as data integration or relation discovery tries to alleviate such quality issues. However, state-of-the-art approaches omit the semantics encoded in the graphs. Semantics provide additional information that may be relevant for the execution of such tasks. We hypothesize that the efficiency of data-driven tasks can be improved by considering semantics during their execution. Thus, we formulate the corresponding research question:

- **Research Question 3**: What is the improvement of using semantic similarity measures on data driven tasks, e.g., to discover relations in a knowledge graph?

To address the third research question, we describe three scenarios where different data-driven tasks are requested and study the enhancement caused by considering semantics on each of them. First, we propose AnnEvol, a methodology to analyze the evolution of an annotation dataset considering semantics. Secondly, we propose KOI, an approach to discover relations between resources in a knowledge graph based on similarity-aware graph partitioning techniques. Finally, one of the described semantic similarity measures is implemented in FuhSen, a similarity-based integration approach for knowledge graphs.

1.3 Running Example

This section describes the example used for describing the works in the state of the art and the different similarity measures that compound the presented framework. The example consists of a knowledge graph of countries and their relations and attributes (see Figure 1.1). The knowledge graph contains a hierarchy formed by three nodes: *World*, *Europe* and *European Union*. Under these three nodes there are several European countries classified, e.g. Norway, Spain or Germany. Apart from the hierarchical relation among the nodes, there are also non-hierarchical or transversal properties. These properties are used to define when two countries share a border (*hasBorderNeighbor*) or when they belong to a different kind of neighborhood, e.g. Schengen area (*hasSchengenNeighbor*). Figure 1.2 shows the hierarchy of the relations contained in Figure 1.1. Finally, countries are also described with attributes. Particularly, there is a relation called *language* defining the language spoken in certain countries. For the sake of clarity some relations between countries as well as some attributes are omitted. In particular examples the knowledge graph may vary adding different kind of entities, e.g., international agreements or organizations, or relations in order to facilitate the explanation and the understanding. Further, depending on the described similarity measures, entities in the graph are considered either RDF classes or instances.

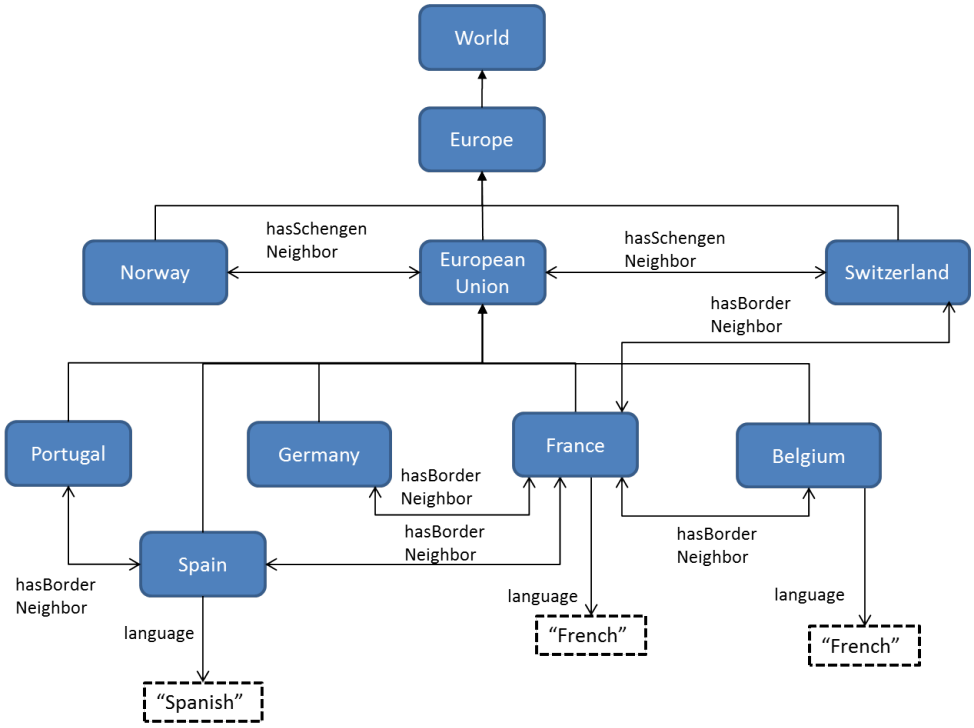


Figure 1.1: Knowledge graph describing European countries and their relations.

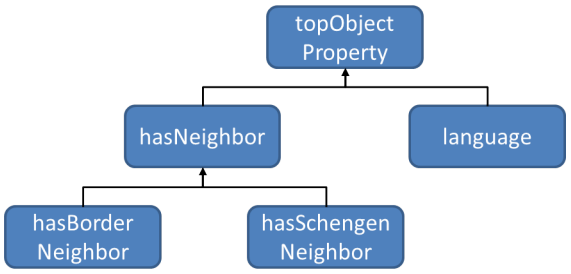


Figure 1.2: Knowledge graph describing the relations used in Figure 1.1 in a hierarchy.

1.4 Previous Publications

The overall research problem is discussed in a doctoral consortium paper [102]:

- Traverso-Ribón, I. (2015). Exploiting semantics from ontologies to enhance accuracy of similarity measures. In *Proceedings of 12th Extended Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015*, pages 795–805

The similarity measures OnSim (4.2.1), IC-OnSim (4.2.2) and GADES (4.3) have been published in the following research papers [107, 104, 105]:

- Traverso-Ribón, I., Vidal, M., and Palma, G. (2015b). Onsim: A similarity measure for determining relatedness between ontology terms. In *Proceedings of 11th International Conference on Data Integration in the Life Sciences, DILS 2015, Los Angeles, CA, USA, July 9-10, 2015*, pages 70–86
- Traverso-Ribón, I. and Vidal, M. (2015). Exploiting information content and semantics to accurately compute similarity of go-based annotated entities. In *Proceedings of IEEE Conference on Computational In-*

telligence in Bioinformatics and Computational Biology, CIBCB 2015, Niagara Falls, ON, Canada, August 12-15, 2015, pages 1–8

- Traverso-Ribón, I., Vidal, M., Kämpgen, B., and Sure-Vetter, Y. (2016b). GADES: A graph-based semantic similarity measure. In *Proceedings of the 12th International Conference on Semantic Systems, SEMANTICS 2016, Leipzig, Germany, September 12-15, 2016*, pages 101–104

The analysis of the evolution of annotation datasets considering semantics with the AnnEvol methodology 5.3 is published in an International per-reviewed conference in the the area of Life Sciences [106]:

- Traverso-Ribón, I., Vidal, M., and Palma, G. (2015a). Annevol: An evolutionary framework to description ontology-based annotations. In *Proceedings of 11th International Conference on Data Integration in the Life Sciences, DILS 2015, Los Angeles, CA, USA, July 9-10, 2015*, pages 87–103

The relation discovery approach KOI is also peer-reviewed and published [103]:

- Traverso-Ribón, I., Palma, G., Flores, A., and Vidal, M. (2016a). Considering semantics on the discovery of relations in knowledge graphs. In *Proceedings of 20th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2016, Bologna, Italy, November 19-23, 2016*, pages 666–680

Finally, the improvement of data integration techniques by considering semantics is published in the following research papers:

- Collarana, D., Galkin, M., Traverso-Ribón, I., Lange, C., Vidal, M., and Auer, S. (2017a). Semantic data integration for knowledge graph construction at query time. In *Proceedings of 11th IEEE International Conference on Semantic Computing, ICSC 2017, San Diego, CA, USA, January 30 - February 1, 2017*, pages 109–116

- Collarana, D., Galkin, M., Traverso-Ribón, I., Vidal, M., Lange, C., and Auer, S. (2017b). Minte: Semantically integrating rdf graphs. In *Proceedings of 7th ACM International Conference on Web Intelligence, Mining and Semantics, WIMS 2017, Amantea, Italy, June 19 - 22, 2017*

- Galkin, M., Collarana, D., Traverso-Ribon, I., Vidal, M.-E., and Auer, S. (2017). Sjoin: A semantic join operator to integrate heterogeneous rdf graphs. In *Proceedings of 28th International Conference on Database and Expert Systems Applications (DEXA), Lyon, France*

1.5 Guide to the Reader

The thesis is structured on six chapters including the introduction, a chapter of foundations, a chapter about related work, two chapters describing the main contributions and a conclusion chapter.

Chapter 2 presents background knowledge necessary to understand the contributions of this work. A formal description of the concepts of metric and knowledge graph are included. Basic definitions of knowledge graph properties, as well as semantic technologies (RDF, OWL, etc.) and several data-driven tasks are also advanced.

Chapter 3 presents an overview of the state of the art. The limitations of current similarity measures are described. Additionally, state-of-the-art graph partitioning techniques and data integration approaches are introduced.

Chapters 4 and 5 are the core chapters of this thesis. Figure 1.3 gives an overview of the contributions described in both chapters. Chapter 4 describes a framework of semantic similarity measures including a theoretical and empirical time complexity analysis. Furthermore, it is formally demonstrated that one of the three measures fulfills the properties of a metric. Chapter 5 shows the benefits of considering semantics in three data-driven tasks for the enhancement of knowledge graph quality issues. First, it is shown how the consideration of semantics can

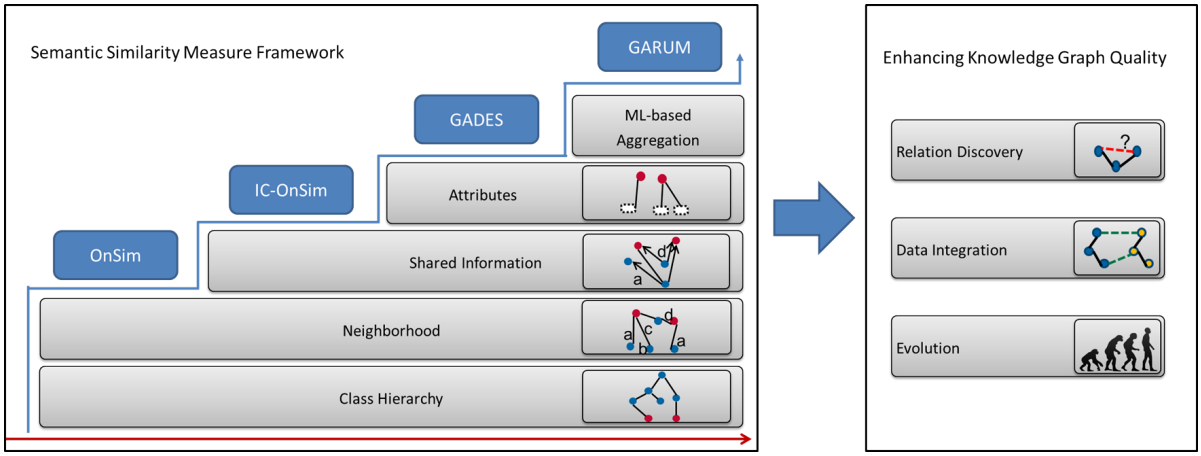


Figure 1.3: Outlook of this PhD thesis.

improve the analysis of the evolution of semantically annotated datasets. Better results are also obtained in the relation discovery task when considering semantics. Finally, one of the semantic similarity measures presented in Chapter 4 is used to improve the effectiveness in the integration of Linked Data datasets.

Chapter 6 presents the conclusions of this thesis and introduce remaining open questions, respectively.

2 Foundations

This chapter presents the foundations of the work reported in this thesis. Section 2.1 includes a formal definition of similarity measure and related concepts, e.g., metric and triangular norm. Section 2.2 introduces the concept of knowledge graph. Section 2.3 defines the graph partitioning problem. Furthermore, Semantic Web technologies utilized to represent and query knowledge graphs, e.g., RDF, OWL, SPARQL are described, as well as related concepts like annotation graphs. A definition of bipartite graph and the 1-1 weighted perfect matching problem is given in Section 2.4. Finally, fuzzy approaches to compare sets are presented in Section 2.5.

2.1 Similarity Measure

This thesis presents a framework of semantic similarity measures able to detect relatedness among knowledge graph entities. Determining relatedness among elements on a set is a dichotomy that relies on two parts: i) Finding the set of relevant features to compute the similarity and their encoding in a feature space, and ii) Providing a suitable metric for a given set of relevant features. Let X be a non-empty set of elements for which an equality relation is defined. Similarity measures compute a degree of coincidence between elements in a set and can be formally defined as functions:

Definition 1. *A similarity measure sim on elements in X is an upper bound, exhaustive, and total function sim* : $X \times X \rightarrow I_s \subset \mathbb{R}$ *with* $|I_s| > 1$ *[9]. I_s represents a set of real numbers containing at least two elements that allow to distinguish similar from dissimilar elements. Defining the range of a similarity measure as I_s is equivalent to defining a minimum and a maximum value. Thus, I_s is upper and lower bounded and the set has both a supremum (sup) and an infimum (inf).*

Values close to 0.0 indicate that the compared objects are dissimilar while values close to the supreme of I_s correspond to very similar objects. There exist three types of similarity measures, 1. distance-based similarity measures, 2. feature-based similarity measures and 3. probabilistic similarity measures [4] . This thesis makes use of distance-based and feature-based similarity measures. Probabilistic similarity measures assume that the comparison among two elements may return different values over the time, i.e., the perception about two elements depends on the time they are observed. In this thesis it is assumed, that changes of perceptions must be reflected in the different versions of the knowledge graph and, therefore, the paradigm of probabilistic similarity measures is not needed. Knowledge graph resources are described through attributes (features); they are connected to each other means paths whose length (distance) can be easily measured.

Distance-based Similarity Measures: Distance-based similarity measures assume that objects can be represented in a multidimensional space. Short distances between objects in that space indicate that the objects are similar. Thus, the similarity is inversely related to the distance. In general, a distance-based similarity measure between two objects x, y can be defined as $sim(x,y) = \phi(d(x,y))$, where d is a distance measure and ϕ a decreasing function, i.e., long distances among objects are transformed to low similarity values.

Feature-based Similarity Measures: Tversky defines the similarity computation as a feature matching process [108]. In general, according to the description of Tversky, a similarity function can be described as a linear combination of their common and distinctive features $sim(x,y) = \alpha g(x \cap y) - \beta g(x-y) - \gamma g(y-x)$, where $g(x \cap y)$

represents the salience of features common for x and y, $g(x-y)$ denotes the salience of the features unique for x, $g(y-x)$ denotes the salience of the features unique for y, and α, β, and γ are constants that are domain-dependent.

For some data-driven tasks, e.g., clustering, only similarity measures that meet the properties of a metric can be used. Otherwise, clusters would not be coherent. Let $\{x,y,z\}$ be a triangle that does not meet the triangle inequality, i.e., $d(x,z) > d(x,y) + d(y,z)$, where d is a distance function. Given that distances $d(x,y)$ and $d(y,z)$ are short, a clustering algorithm would try to assign to the same cluster for x, y, and z. However, the triangle does not meet the triangle inequality and the distance $d(x,z)$ can be of any length. Thus, assigning the same cluster to x and z may be penalized by the clustering algorithm, so an incoherent situation would be reached [6].

Metrics

Definition 2. *A metric or distance function on elements in a set X is a function* $d : X \times X \rightarrow \mathbb{R}_0^+$ *that fulfills the following conditions:*

- *Non-negativity: The distance among two elements* $a, b \in X$ *cannot be negative* $d(a,b) \geq 0.0$.
- *Identity: The distance among two elements* $a, b \in X$ *is 0.0 iff they are the same element* $d(a,b) = 0.0 \iff a = b$.
- *Symmetry: The distance between two elements* $a, b \in X$ *is symmetric* $d(a,b) = d(b,a)$
- *Triangle inequality: The distance between two elements* $a, c \in X$, $d(a,c)$ *is minimal, i.e., for all element* $b \in X$ *the expression* $d(a,c) \leq d(a,b) + d(b,c)$ *applies.*

Given that this work focuses on similarity functions, the properties of a metric are transformed for similarity measures as follows:

Definition 3. *A similarity measure sim is considered a metric if the following properties are fulfilled:*

- *Limited range:* $sim(x,y) \leq sup\ I_s$, *where sup represents the supremum of the set* I_s.
- *Reflexivity:* $sim(x,y) = sup\ I_s \iff x = y$
- *Symmetry:* $sim(x,y) = sim(y,x)$
- *Triangle inequality:* $sim(x,z) \geq sim(x,y) + sim(y,z) - sup\ I_s$

Usually, similarity measures determine the similarity between two objects aggregating similarity values for several features. Thus, it is necessary to combine the different similarity values to obtain an aggregated value. There are different aggregation strategies, e.g., the arithmetic mean. Other aggregation strategies allow for weighting the different similarity values to be combined according to their relevance. Features with a high weight have more influence in the final similarity value. However, not all aggregation strategies allow for the satisfaction of the metric properties. Triangular norms are aggregation functions that ensure the fulfillment of such properties.

2.1.1 Triangular Norm

Triangular norms have been used mainly in the fields of probabilistic metric spaces and in fuzzy logic. Menger introduces the first definition of the triangular norm concept [62]. Menger defines a metric space where the distance between two elements would be represented by a probability distribution. These metric spaces are nowadays named as probabilistic metric spaces. Triangular norms are used in this setting to generalize the triangle inequality of classic metric spaces. Furthermore, triangular norms have been used in fuzzy logic as a way of computing the intersection among two fuzzy sets. In this thesis, triangular norms are utilized to aggregate multiple metrics preserving the triangular inequality property. Formally, a triangular norm (t-norm) is a binary operation $T : [0,1] \times [0,1] \rightarrow [0,1]$ with the following algebraic properties [46]:

Definition 4 (Triangular Norm)**.** *A triangular norm T is a binary operation in the unit space $T : [0,1] \times [0,1] \to [0,1]$, such that for all $x, y, z \in [0,1]$ the following axioms hold:*

$$
\begin{aligned}
&T(a,b) = T(b,a) && \text{(commutativity)}\\
&T(a,b) \leq T(c,d) \text{ if } a \leq c \text{ and } b \leq d && \text{(monotonicity)}\\
&T(a,T(b,c)) = T(T(a,b),c) && \text{(associativity)}\\
&T(a,1) = a && \text{(identity element)}
\end{aligned}
$$

There exist a number of different t-norms classified in different families. Given that it is not the focus of this work to give an extended view of t-norms, only the basic ones are listed. The basic known t-norms are the minimum T_M, the product T_P, the Lukasiewicz t-norm T_L, and the drastic product T_D and which are described as follows:

- $T_M(x,y) = \min(x,y)$ (minimum)
- $T_P(x,y) = x * y$ (product)
- $T_L(x,y) = \max(x+y-1,0)$ (Lukasiewicz)
- $T_D(x,y) = \begin{cases} 0, & \text{if}(x,y) \in [0,1)^2 \\ min(x,y), & \text{otherwise} \end{cases}$ (drastic product)

2.2 Knowledge Graphs

Knowledge graphs are a knowledge representation method that belongs to the category of semantic networks. A knowledge graph can be defined formally as follows:

Definition 5. *Given a set of nodes V, a set of edges E and a set of property labels L, a knowledge graph G is defined as $G = (V,E,L)$. Edges in E correspond to triples of the form (v_1, r, v_2), where $v_1, v_2 \in V$ are entities in the graph, and $r \in L$ is a property label.*

Nodes in a knowledge graph represent concepts of the real world, while edges represent relations between these concepts. Property labels describe the meaning or type of the relation and make the graph understandable for humans. According to their meaning, property labels are categorized into two classes: taxonomic and transversal.

Taxonomic properties In the context of knowledge graphs, a taxonomy is a hierarchical classification of the concepts. In a taxonomy, nodes are connected through subsumption relations. A concept A subsumes B if every element of type B is also of type A. Taxonomic properties are transitive, i.e., if A subsumes B and B subsumes C, then A also subsumes C. Thus, taxonomic properties induce a partial order in the knowledge graph.

Transversal properties Transversal or horizontal properties do not have to induce a partial order among the related concepts, i.e., they express semantic non-taxonomic properties. The n-hop neighborhood of a node v is the set of nodes N that can be reached from v through transversal properties in at most n hops. In some kind of graphs like social network graphs, the 1-hop neighborhood is also named as *Ego-Network*.

Following concepts and technologies relevant to knowledge graphs are described. Sections 2.2.1 and 2.2.2 introduce the RDF description framework and its extension RDFS together with the Web Ontology Language OWL respectively. Next, in Section 2.2.3 the W3C definition of annotation graph is presented. Section 2.2.4 describes the knowledge base concepts and names some of the currently most prominent knowledge bases. Finally, Section 2.2.5 gives a short introduction into SPARQL, a query language for RDF data.

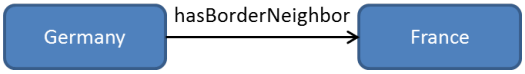


Figure 2.1: Example of triple extracted from DBpedia.

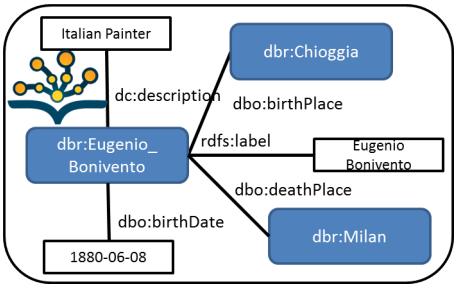


Figure 2.2: Portion of the RDF molecule of Eugenio Bonivento in DBpedia [13].

2.2.1 RDF: Resource Description Framework

The Resource Description Framework (RDF) is a metadata data model specified by the World Wide Web Consortium [111] to describe web resources similar to classical conceptual modeling approaches as class diagrams or semantic networks. Thus, knowledge modeled with RDF can be represented in a graph-based way. Hence, an RDF graph is composed of nodes, literals, and predicates, that label the edges that interconnect the nodes. Nodes and predicates are unambiguously identified through Internationalized Resource Identifiers (IRIs) [19]. Each IRI identifies exactly one resource, e.g., one RDF node or predicate, in the whole Web. However, an RDF node or predicate can be identified through multiple IRIs. Edges in RDF are represented as triples with the form (*subject*, *predicate*, *object*), where *subject* is the origin of the edge, *predicate* corresponds to the label of the edge and *object* is the target. The object of the triple can be another node or a literal. The next triple is represented in Figure 2.1:

Subject: `http://url.org/Germany`

Predicate: `http://url.org/hasBorderNeighbor`

Object: `http://url.org/France`

RDF Molecule: Often users are only interested in just a subset of the resources described in the knowledge graph. An RDF molecule of a resource in an RDF knowledge graph collects all the information regarding this resource in the knowledge graph, i.e., all the statements with the relevant resource as subject. Figure 2.2 contains an example of an RDF molecule. Formally, an RDF molecule is defined as follows:

Definition 6. *Let G be a knowledge graph represented in RDF. An RDF molecule [21] is a subgraph of G $M \subseteq G$ such that all triples in M share the same subject, i.e., $\forall (s_i, p_i, o_i), (s_j, p_j, o_j) \in M \mid s_i = s_j$.*

An RDF molecule can be represented with a tuple $M = (s, T)$, where s is the resource subject of all triples in the molecule and T is a set of pairs (p, o) such that $\forall (p, o) \in T$ the triple $(s, p, o) \in G$ is contained in the graph G. T can be also identified as the tail of the molecule.

2.2.2 RDF Schema and OWL

RDF Schema [11] extends the RDF data model enabling users to express more complex statements and provides mechanisms to introduce more structure for knowledge represented in RDF. The main concepts and predicates introduced in RDFS are `rdfs:Class`, `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:domain` and `rdfs:range`.

rdfs:Class The rdfs:Class concept allows user to group resources that belong to a certain class. These resources are called instances of the class. Hence, classes like Person, Protein, or Book can be found in the publicly available knowledge bases like DBpedia [5], WikiData [109] or YAGO [59].

rdfs:subClassOf The rdfs:subClassOf predicate enables one class to be subsumed by another one. This predicate allows for the definition of a class hierarchy, one of the relevant features to compute the similarity among two resources. For example, the DBpedia ontology describes an aircraft as a mean of transportation with the triple dbo:Aircraft rdfs:subClassOf dbo:meanOfTransportation.

rdfs:subPropertyOf Similarly to rdfs:subClassOf, the rdfs:subPropertyOf predicate enables to define a property hierarchy, so a property is subsumed by another one. For example, the DBpedia ontology defines that a bronze medalist is a kind of medalist dbo:bronzeMedalist rdfs:subPropertyOf dbo:Medalist.

rdfs:domain The rdfs:domain property enables users to define the domain of a certain rdf:Property, i.e., to which class must the resource belong that acts as subject of the corresponding triple.

rdfs:range The rdfs:range property enables users to define the range of a certain rdf:Property, i.e., to which class must the resource belong that acts as object of the corresponding triple.

The Web Ontology Language (OWL) [61] extends the RDF Schema data model with even more expressiveness. Thus, the richer vocabulary of OWL allows to define the cardinality of the properties (owl:maxCardinality), logical operators as the union of two classes (owl:unionOf), or to specify that a certain predicate is transitive (owl:TransitiveProperty). Next, the most relevant characteristics of OWL for this thesis are described.

Complex class expressions OWL allows for the definition of a certain class subsumed by a class expression, forcing objects of this class to meet a certain property. For example, a class representing persons that are parents may be expressed as: ex:Parent rdfs:subClassOf (ex:Person and ex:isParentOf some ex:Person). Thus, every instance of the class ex:Parent must be related through the property ex:isParentOf with another instance of the class ex:Person. Nevertheless, because of the Open World Assumption, this instance may not be defined yet.

Transitivity Transitivity enables OWL reasoners to infer new triples in a knowledge graph. Particularly, when a property ex:Property is declared transitive, then the triple (ex:A, ex:Property, ex:C) is inferred from the triples (ex:A, ex:Property, ex:B) and (ex:B, ex:Property, ex:C). For example, if a person A is taller than B and B is taller than C, then the person A has to be also taller than the person C.

Symmetry An RDF property can be defined as symmetric in OWL. If a property ex:Property is declared symmetric, an OWL reasoner can infer the triple (ex:A, ex:Property, ex:B) from the existence of the triple (ex:B, ex:Property, ex:A). For example, if person A is a colleague of person B, then the person B has to be also colleague of the person A.

Due to the high expressivity introduced in OWL, the execution of a reasoner on an OWL ontology is costly in terms of computation. The computational complexity of classifying an OWL ontology has worst case complexity of 2NEXP-Time [45]. OWL2 introduces three OWL profiles [65] to reduce the complexity of the different tasks performed by a reasoner depending on the aim of the ontology. A profile is a portion of the OWL2 data model.

OWL2 EL is useful for ontologies with a large number of properties or classes and allows for performing basic reasoning tasks as classification or axiom entailment in polynomial time with respect to the size of the ontology. The EL acronym refers to the EL family of description logics.

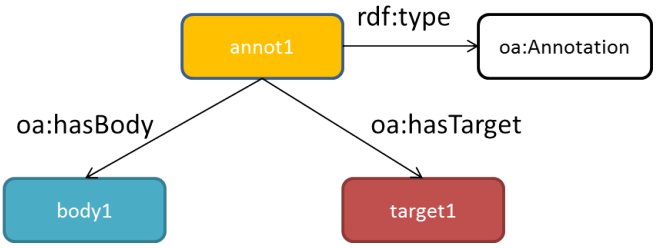


Figure 2.3: Basic Annotation Model [87]

OWL2 QL is suitable for ontologies with a large amount of instances. The reasoning task improved is query answering, which can be performed in LOGSPACE with respect to the size of assertions in the ontology. The profile is based on the DL-Lite family of description logics.

OWL2 RL optimizes the execution of most of the reasoning tasks losing the minimum expressivity. The OWL2 RL profile comprises the subset of OWL2 that can be implemented by means of rule-based technologies. Thus, more complex reasoning tasks as class expression satisfiability can be performed in polynomial time.

2.2.3 Annotation Graph

Controlled vocabularies enable humans to describe precisely real world entities. Though such vocabularies are not as expressive as textual descriptions, they allow to reduce the ambiguity of textual descriptions. However, computers are not able to understand how the different vocabulary terms are related. Ontologies can be used to describe the properties of the terms in a controlled vocabulary and their relations. Thus, computers can use the knowledge encoded in the ontology to understand the semantics of each term. An example of annotation graph can be found in UniProt [100], where proteins are described with Gene Ontology terms.

The World Wide Web Consortium (W3C) promotes the annotation of web resources with ontologies. One of their initiatives is the standardization of the annotation process to make annotations easily shareable. Hence, they published in 2013 the Open Annotation Data Model [87]. According to this specification, an annotation is a set of resources including a target and a body. The body is related to the target, i.e., the body describes a property of the target. Figure 2.3 shows an example of the basic annotation model proposed by the W3C.

2.2.4 Knowledge Bases

Semantic Web and Linked Data initiatives have fostered the representation of knowledge in RDF knowledge graphs. Thus, different knowledge bases like DBpedia [5], Wikidata [109] or UniProt [100] were created to represent knowledge from different domains. These knowledge bases are used and referenced in this work. A brief introduction of these knowledge bases is provided as follows.

DBpedia DBpedia is a general domain knowledge base that was created in 2007 from a research project conducted by Auer et al. [5]. The motivation of DBpedia is to take advantage of the knowledge available in Wikipedia in the form of infoboxes or tables and build a knowledge base from it. As the knowledge in Wikipedia is collaboratively created, the mappings between table fields and the DBpedia ontology are maintained and curated by the community. Further, DBpedia also implements methods to automatically extract triples from text. However, the precision of these methods is lower than the table extractors; therefore, false statements may be generated. DBpedia offers two versions of their dataset. The first one includes only triples extracted from the Wikipedia tables, i.e., high quality triples. The second one also includes triples extracted from free text. Today DBpedia is one of the

main knowledge bases on the Linked Open Data Cloud with more than 9.5 billion triples and 127 languages[1]. DBpedia is not isolated, but connected to other knowledge bases like Wikidata or Freebase, facilitating the collection of the information regarding a certain resource.

Wikidata Wikidata is another general domain knowledge base initiated by Wikimedia foundation in 2012. Wikidata aims to offer a central repository for statements that are consumed by wikis. Unlike DBpedia, where the knowledge is extracted from Wikipedia, the knowledge in Wikidata is curated and described by humans. Today Wikidata contains more than 66 millions of triples in more than 50 languages [99].

UniProt The Universal Protein Resource (UniProt) is a knowledge base in the protein domain [100]. UniProt describes proteins including structural properties as their sequences and functional properties. Functional properties are described with annotations from the Gene Ontology (GO). UniProt is divided into two knowledge bases called SwissProt and TrEMBL. Proteins in SwissProt are described with manually curated annotations, i.e., annotations whose referred properties have been empirically proved or found in the literature. However, TrEMBL also includes computationally generated annotations. These annotations are automatically generated and has not been experimentally confirmed. UniProt contains more than 15 billions triples describing more than 3 billions proteins[2].

2.2.5 SPARQL

SPARQL is the W3C Recommendation [101] for querying and manipulating RDF data on the web or an RDF store. The relevant part of this recommendation for this work is the SPARQL query language [34]. A SPARQL query consists of up to five parts:

Prefix Declaration A list of URI prefixes to avoid writing complete URIs in the query.

Dataset Clause Similarly to SQL databases, where the user specifies the schema to be used, in the dataset clause is specified which graph is going to be queried.

Result Clause In this clause the type of query (SELECT, ASK, CONSTRUCT or DESCRIBE) and the variables to return are specified.

Query Clause The query clause contains the patterns that have to be matched in the graph. Resources fulfilling the specified patterns will be associated with the corresponding variables in the result clause.

Solution Modifiers The results of the queries can be paginated, ordered or sliced.

The result clause allows for four types of SPARQL queries:

SELECT A mapping between graph resources and variables is returned.

ASK A Boolean value indicating if some resource matches the specified pattern in the graph is returned.

CONSTRUCT This type of query returns an RDF graph built by replacing variables in a set of triple templates, which indicates how the graph has to be constructed.

DESCRIBE The describe query returns an RDF graph describing the resources that match the specified patterns.

In this thesis, SPARQL is used for describing queries whose evaluation allows for collecting data from different knowledge bases like DBpedia or Wikidata; further, SPARQL is utilized to define constraints during the relation discovery task. Therefore, the SELECT and the ASK queries are the most relevant for this work. Figures 2.4 and 2.5 show examples of ASK and SELECT queries.

[1] Source `http://wiki.dbpedia.org/dbpedia-version-2016-04`, retrieved on 2017-04-18.
[2] Source `http://sparql.uniprot.org/.well-known/void`, retrieved on 2017-04-18

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
ASK  {
  { SELECT ?competition
      WHERE {?competition dbo:goldMedalist dbr:Ian_Thorpe}
  }
}
```

Figure 2.4: SPARQL ASK query answering if Ian Thorpe won a gold medal in any competition.

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?competition
      WHERE {?competition dbo:goldMedalist dbr:Ian_Thorpe}
  }
```

Figure 2.5: SPARQL SELECT query returning the competitions where Ian Thorpe obtained the gold medal.

2.3 Graph Partitioning

Graphs are used in several domains as way of representing information. In the era of Big Data, these graphs can be huge and its atomic processing may be impossible in some cases. Following the divide-and-conquer strategy, computer scientists divide huge graphs into smaller cuts that can be easier processed. The problem is formally described as follows:

Definition 7. *Let $G = (V,E)$ be a graph with nodes V and edges E. The partition of G can be expressed as $P(G) = \{p_1, p_2, ..., p_n\}$ where:*

- *each part in the partition is a subset of vertices (or edges) $\forall i \in \{1,2,...,n\}, p_i \in P(G) \mid p_i \subseteq V$*
- *every vertex $v \in V$ is included in one and only one part $p \in P(G)$. Formally, $\forall i, j \in \{1,2,...,n\}$, if $i \neq j$ and $p_i, p_j \in P(G)$, then $p_i \cap p_j = \emptyset$ and $V = \bigcup_{p \in P(G)} p$.*

A graph can be partitioned in a finite number of ways. In order to decide which partition is better, an objective function is defined. Thus, a graph partitioning algorithm will return the partition with the best value for the objective function. A common objective function is the graph conductance. The conductance of a part p of a graph G is defined as:

$$\phi(p) = \frac{\sum_{v \in p, w \in \overline{p}} a_{vw}}{\min(a(p), a(\overline{p}))},$$

where v and w represents nodes in the graph, a_{vw} represents the entries in the adjacency matrix and $a(p) = \sum_{v \in p, w \in V} a_{vw}$. Using conductance as object function, the graph partitioning algorithm will find a partition of the graph whose parts are densely connected internally, but with few connections to other parts.

Graph partitioning algorithms are used in this thesis to identify graph portions including highly similar resources; these algorithms are used in the relation discovery and annotation evolution analysis tasks (Sections 5.2 and 5.3).

SemEP

Palma et al. define semEP [71], a graph partitioning algorithm that finds the minimal partition of a bipartite graph $BG = (U \cup V, WE)$, such that the aggregated density of each part is maximal. Thus, the objective function of semEP is not the conductance, but the aggregated density of the found partition, which is defined as $\frac{\sum_{p \in P(BG)} cDensity(p)}{|P(BG)|}$. The density of each part *cDensity* is computed according to a similarity measure defined for nodes in the bipartite graph $cDensity(p) = \frac{\sum_{(u,v) \in p} \text{sim}(u,v)}{|p|}$, where $u \in U$, $v \in V$ and $(u,v) \in WE$.

In this thesis semEP is used during the relation discovery task in KOI, as well as during the analysis of the evolution of annotation datasets in AnnEvol. KOI discovers relations among highly similar entities. SemEP is used

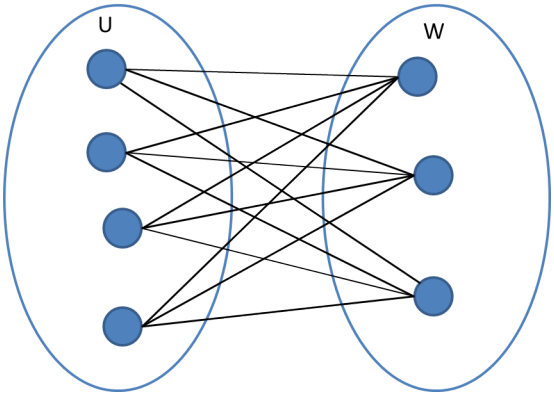


Figure 2.6: Complete Bipartite Graph

by KOI to detect portions of a graph containing highly similar entities. AnnEvol allows to analyze the evolution of an annotation graph from a set-wise perspective. SemEP is used by AnnEvol to compute AnnSig, a similarity measure that detects relatedness between portions of the annotation graphs instead of between annotations pairs.

2.4 Bipartite Graph

Bipartite graphs have the property that their vertices can be separated into two parts, such that no pair of vertices in the same part are connected with any edge. This property makes bipartite graphs suitable for applications where it is necessary to represent the interaction among two types of entities like jobs and workers, football players, and clubs. Figure 2.6 shows an example of a bipartite graph and a formal definition is as follows:

Definition 8 (Bipartite Graph)**.** *A bipartite graph or bigraph is a graph $G = (V,E)$ whose vertices can be split into two disjoint sets $V = U \cup W$, $U \cap W = \emptyset$ such that every edge in E connects a vertex from U to a vertex in W. This constraint is expressed formally as follows: $\forall (u,w) \in E \rightarrow u \in U \wedge w \in W$.*

In this thesis, bipartite graphs are used for comparing ego-networks or 1-hop neighborhood of entities in a knowledge graph. Further, bipartite graphs are utilized to integrate RDF molecules described in different Linked Data datasets with different Linked Data vocabularies. In bipartite graphs, edges are weighted with the corresponding similarity values. Computing the similarity value between the two sets of vertices requires of an aggregation strategy. There are different ways of aggregating these values to obtain a similarity value among the two sets. A naive strategy would be to compute the average of all the weights. However, more intelligent strategies may be desirable to maximize the accuracy of the aggregated similarity value. One of these strategies is the 1-1 Maximum Weighted Perfect Matching, which allows to identify which pair of elements in the compared sets contributes to a greater extent to the final similarity value.

2.4.1 1-1 Maximum Weighted Perfect Matching

One of the most prominent applications of bipartite graphs is the assignment problem. The assignment problem is a combinatorial optimization problem consisting of finding the best matching on a bipartite graph according to a certain optimization criteria, e.g., maximization or minimization, with respect to the weights of the edges in this graph. The problem is formally defined as follows:

Definition 9 (1-1 Maximum Weighted Perfect Matching)**.** *Let $G = (U \cup V, WE)$ be a bipartite graph, where U and V are two sets of graph nodes and WE is a set of edges weighted with real values among nodes in U and V. A 1-1 matching is a set of weighted edges $ME \subseteq WE$, such that none of them share a vertex, i.e., $\forall (u,v),(x,y) \in ME, \{u,v\} \cap \{x,y\} = \emptyset$. Figure 2.7 contains an example of 1-1 bipartite graph matching. The 1-1* maximum weight

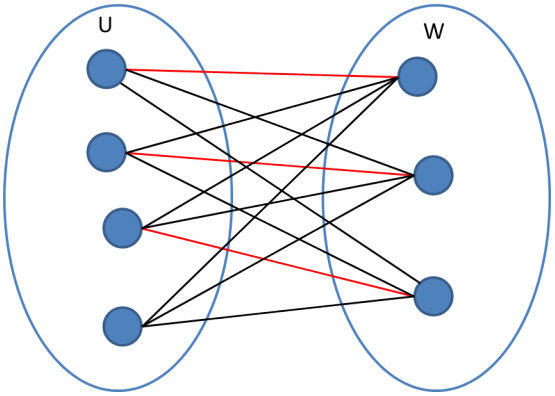


Figure 2.7: Bipartite Graph. Red edges represent the computed 1-1 matching.

bipartite matching *[90] between the edges in WE is represented as another bipartite graph* $MWBG = (U \cup V, WE_r)$ *and has the following properties:*

- $\text{WE}_r \subseteq$ WE, *i.e.,* MWBG *is a sub-graph of* G.
- *the sum of the weights of the edges in* WE_r *is maximized, i.e.,* $max \sum_{(u,v) \in WE_r} sim(u,v)$, *where* $u \in U$ *and* $v \in V$.
- *for each node n in* $U \cup V$ *there is only one incident edge e in* WE_r, *i.e.,*
  - $\forall\, u \in U$, *there is only one v in V, such that* $e = (u,v) \in \text{WE}_r$, *and*
  - $\forall\, w \in V$, *there is only one x in U, such that* $e = (x,w) \in \text{WE}_r$.

There are several algorithms that solve the assignment problem in polynomial time, being the Hungarian algorithm [50] the most popular with a computational complexity of $O(n^3)$.

The 1-1 maximum weighted perfect matching by the diffrent defined similarity measures to compare neighborhoods of ontology terms or knowledge graph entities (Chapter 4); further, it is utilized during the knowledge graph integration task to determine which RDF molecules should be integrated (Section 5.4).

2.5 Comparison of Sets

Semantic similarity measures enable the comparison of real-world entities annotated with ontology terms and knowledge graph resources. Both types of comparisons require the computation of similarity values among sets. These sets correspond either to the sets of ontology terms used to annotate the real-world entities or the 1-hop neighborhoods of the knowledge graph resources. Set similarity values are often computed as an aggregation of the similarity values of their single elements. Thus, a method to aggregate ontology term or knowledge graph resource similarity values is needed. Next, an overview of suitable set comparison methods is given:

Jaccard Jaccard computes the similarity between two sets A and B according to the amount of elements in their intersection with respect to their union $\text{Jaccard}(A,B) = \frac{|A \cap B|}{|A \cup B|}$. Jaccard does not compute any similarity value among the elements in the sets; however, it identifies when those elements are or are not the same.

Average A method that considers the similarity values among the elements is the average. Given the bipartite graph $G = (A \cup B, WE)$ with edges in WE weighted according to a similarity measure, the similarity value among A and B is defined as $\frac{\sum_{(x_i,y_j) \in WE_r} \text{sim}(x_i,y_j)}{|A_1| \cdot |A_2|}$. This approach does not meet the metric properties and all comparisons are equally considered.

AnnSim Palma et al [70] define AnnSim, a similarity measure to compare entities annotated with ontology terms. Given two entities e_1 and e_2 annotated with sets of ontology terms A and B, the semantic similarity measure AnnSim(e_1,e_2) determines relatedness between e_1 and e_2 according to the 1-1 maximum weighted perfect matching of A_1 and A_2 $MWBG = (A \cup B, WE_r)$ where edges in $(x_i, y_j) \in WE_r$ are weighted with the similarity value $\text{sim}(x_i, y_j)$. Values of *AnnSim(e_1,e_2)* are computed using the expression:

$$\text{AnnSim}(e_1, e_2) = \frac{2 * \sum_{(x_i,y_j) \in WE_r} \text{sim}(x_i, y_j)}{|A| + |B|}$$

.

AnnSig AnnSig is also a similarity measure for comparing entities annotated with ontology terms. AnnSig is defined as the aggregated density of the partitioning found by semEP. Given a bipartite graph $BG = (U \cup V, WE)$, AnnSig is defined as:

$$\text{AnnSig}(U, V) = \frac{\sum_{p \in P} cDensity(p)}{|P|}$$

where P is the minimal partitioning of the graph according to semEP and $cDensity(p)$ is defined as $cDensity(p) = \frac{\sum_{(u,v) \in p} D_{tax}(u,v)}{|p|}$.

Fujita Fujita [25] presents a distance function for comparing sets based on average distances between their elements. The presented function meets the metric properties, which makes it suitable for clustering algorithms. The distance function is defined with the following formula:

$$f(A, B) = \frac{1}{|A \cup B||A|} \sum_{a \in A} \sum_{b \in B \setminus A} d(a, b) + \frac{1}{|A \cup B||B|} \sum_{b \in B} \sum_{a \in A \setminus B} d(a, b)$$

3 Related Work

In this chapter an overview of the state of the art is provided. First, in Section 3.1, similarity measures based only on the hierarchy to determine similarity values are presented. In Section 3.2 similarity measures based on the amount and length of paths among nodes in a knowledge graph are introduced. Further, Section 3.3 describes similarity measures that compute similarity values based on Information Content coefficients. Section 3.4 introduces lexical and semantic string similarity measures. Section 3.5 introduces similarity measures that combine the information of several resource characteristics to compute similarity values. Finally, Sections 3.6 and 3.7 present the state-of-the-art works in the area of relation discovery and evolution of annotation datasets.

3.1 Taxonomic Similarity Measures

Taxonomic similarity measures consider only taxonomic or hierarchical properties during the computation of the similarity between two nodes in a knowledge graph. In this section D_{ps} [75] and D_{tax} [10] are described as the most representative measures inside this category. Actually, both measures compute the distance among two nodes, but they can be transformed to a similarity measure by calculating the inverse. For the sake of clarity both distance measures are transformed to similarity measures and presented in this chapter. Both measures are based on the distance of the compared nodes to their *Lowest Common Ancestor* (LCA), i.e., the deepest common ancestor. The depth of a node is measured as the length of the shortest path from the node to the root of the hierarchy.

Knowledge graphs encode information not only in hierarchical properties, but also in transversal properties and literals. Taxonomic similarity measures omit the information encoded in these other resource characteristics and, therefore, return inaccurate similarity values, e.g., in knowledge graphs with flat hierarchies. In Figure 3.1 hierarchical properties (*rdfs:subClassOf* and *rdf:type*) classify countries according to their continent. Given that transversal properties (e.g., *hasNeighbor*) are omitted, taxonomic similarity measures will fail detecting that Belgium and France are more similar than Belgium and Portugal. Belgium and France share Germany as neighbor. However, Portugal and Belgium have nothing in common apart from being in the European Union.

3.1.1 D_{ps}

D_{ps} [75] computes the similarity between two nodes in a knowledge graph $G(V,E,L)$ (see Definition 5) and is defined as follows:

$$D_{ps}(x,y) = \frac{d(root, lca(x,y))}{d(x,lca(x,y)) + d(y,lca(x,y)) + d(root,lca(x,y))}$$

where d represents the distance in number of hops between two nodes and $lca(x,y)$ corresponds to the lowest common ancestor among x and y. Nodes with close LCA returns values close to 1, while values close to 0 correspond to nodes with a distant LCA.

3.1.2 D_{tax}

D_{tax} [10] computes the similarity between two nodes in a knowledge graph $G(V,E,L)$ considering only taxonomic or hierarchical edges. D_{tax} is defined as follows:

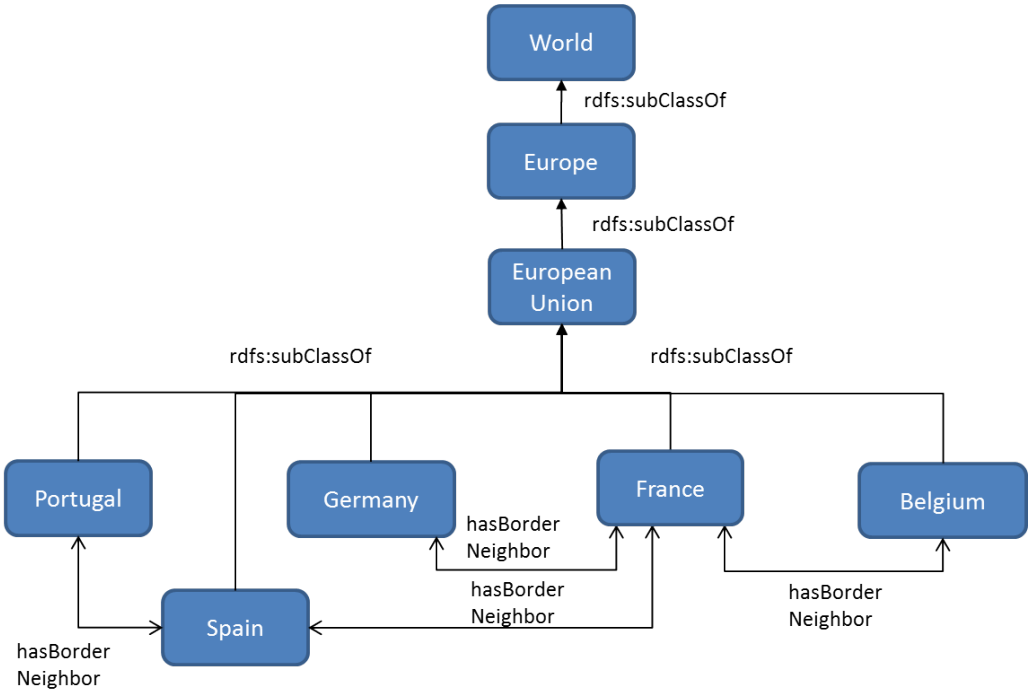


Figure 3.1: Portion of a knowledge graph that describe countries in the world according to their continent and neighbors.

$$D_{tax}(x,y) = 1 - \frac{d(x,lca(x,y)) + d(y,lca(x,y))}{d(root,x) + d(root,y)}$$

where d represents the distance in number of hops between two nodes and $lca(x,y)$ corresponds to the lowest common ancestor among x and y. D_{tax} assigns lower values of distance to pairs of nodes that are at greater depth in the taxonomy and closer to the lowest common ancestor.

3.2 Path Similarity Measures

Path similarity measures compute similarity values based on the distance and the amount of paths between two nodes in a knowledge graph. Usually, these measures do not distinguish among the types of relations that connect the different nodes. Nevertheless, some of them allow the user to define which edges or sequence of edges are relevant when determining similarity values and only consider those edges during the computation. These sequence of edges are called meta-paths. Figure 3.2 describes countries according to their partners and neighbors in a knowledge graph. Path similarity measures that do not distinguish between relation types will fail detecting that Belgium and Germany are more similar than Belgium and Norway. Path similarity measures consider *hasBorderNeighbor* and *hasSchengenNeighbor* properties as they were the same, i.e., as they had the same semantics.

3.2.1 PathSim

PathSim [97] is a similarity measure for heterogeneous information network, i.e., information networks involving multiple types of nodes and relations (see Figure 3.3). Previous similarity measures were designed for homogeneous information networks and do not consider the semantics implicit in relation and node types. To overcome this problem, Sun et al. introduce the concept of meta-path. A meta-path is path consisting of a sequence of relations present in the information network. In the case of PathSim, the meta-path must start and end in nodes of the same type. For example, in Figure 3.3 a meta-path may be *sign o isSigned*. According to PathSim, the similarity between two nodes is directly proportional to the amount of paths that meet the meta-path description among them.

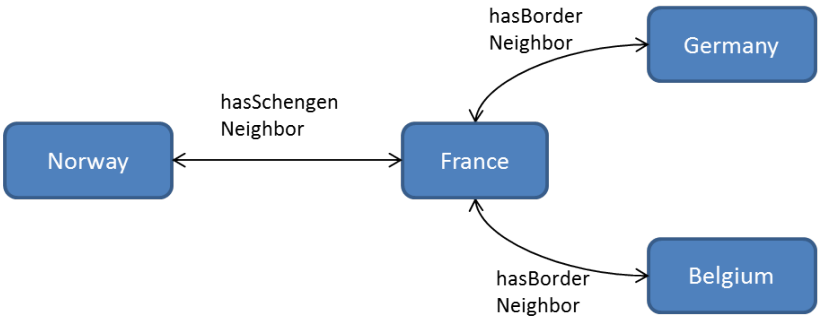


Figure 3.2: Portion of a knowledge graph that describe countries in the world according to their partners and neighbors.

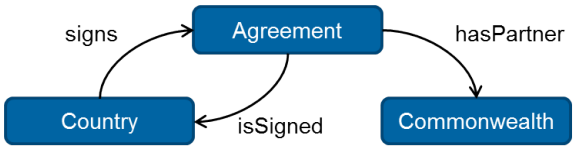


Figure 3.3: Information Network about agreements among countries. A country signs an agreement. The agreement can have a commonwealth as partner, e.g., the European Union.

In Figure 3.3 with the above meta-path definition, the similarity among two actors is proportional to the amount of films they participate. The definition of PathSim is as follows:

$$\text{PathSim}(x,y) = \frac{2|\text{Paths}(x,y)|}{|\text{Paths}(x,x) + |\text{Paths}(y,y)|},$$

where x and y are nodes of the same type in the information network; $\text{Paths}(x,y)$ returns all the different paths that follows the meta-path definition between x and y.

3.2.2 HeteSim

In a conventional similarity search, the comparisons are performed between objects of the same type. However it is necessary to calculate the relatedness between objects of different types. For example, finding the most relevant concepts to a person, i.e., friends, interests, work, or education. HeteSim [92] is as PathSim dependent of a meta-path. The meta-path gives an implicit definition about what is considered similar. In the case of HeteSim the ends of the meta-path must not be nodes of the same type. For example, in Figure 3.3, *signs o hasPartner* is a valid meta-path for HeteSim. Given a meta-path $P = R_1 o R_2 ... R_l$ and two nodes x and y, if $x = y$ then the returned value is 1, otherwise HeteSim is defined recursively as follows:

$$\text{HeteSim}(x,y|R_1 o R_2 o ... R_l) = \frac{\sum_{i=1}^{O(x,R_1)} \sum_{j=1}^{I(x,R_l)} \text{HeteSim}(O_i(x,R_1), I_j(y,R_l)|R_2 ... R_{l-1})}{O(x,R_1) + I(y,R_l)},$$

where $O(x,R_1)$ and $I(y,R_l)$ correspond to the nodes that can be reached from x through an R_1 path and the nodes from which y can be reached through an R_l path. The intuition behind HeteSim is that the similarity between two nodes based on a certain meta-path P is proportional to the probability that x and y meet at the same node when x follows along the path and y against the path.

3.2.3 SimRank

Jeh et al. [41] present SimRank, a domain independent similarity measure. The intuition behind SimRank is that similar objects (nodes) tend to be related to similar objects, i.e., two nodes are similar if their neighbors are similar.

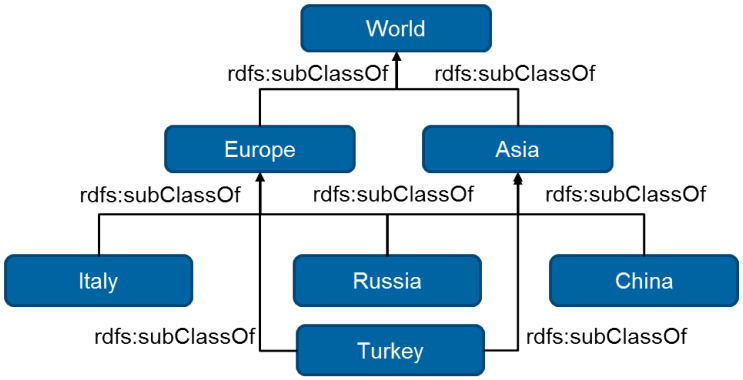


Figure 3.4: Example of taxonomy with multiple inheritance

Unlike above introduced similarity measures, SimRank considers all edge labels included in the knowledge graph during the computation of the similarity. Jeh et al. define SimRank as follows:

$$\text{SimRank}(x,y) = \begin{cases} 1 & x = y \\ \frac{C}{|I(x)||I(y)|} \sum_{i=1}^{|I(x)|} \sum_{j=1}^{|I(y)|} \text{SimRank}(I_i(x), I_j(y)) & \text{otherwise} \end{cases},$$

where C is a value in $[0,1]$ that represents the decay level and $I(x)$ the set of in-neighbors of x. Thus, closer neighbors contribute more than further nodes to the similarity value.

3.3 Information Content Similarity Measures

Taxonomic similarity measures return high similarity values for nodes that are deep and close in the taxonomy. If two nodes are close in a taxonomy, they have a close common ancestor and therefore share a lot of properties. If this common ancestor is deep in the ontology, it is considered very specific too. This means that the two compared nodes share a lot of properties and that these properties are very specific. Thus, the returned similarity is greater. However, the taxonomy may not reflect correctly the abstraction or specificity of a concept (node) in a taxonomy. Information Content based similarity measures consider that the depth in the taxonomy does not reflect the specificity degree, but the use of the nodes in a corpus. These measures follow the Information Content definition provided by Ross [85], that quantifies the Information Content of a concept c in a corpus as $IC(c) = -\log p(c)$, where $p(c)$ represents the likelihood of finding the concept c in the corpus. Based on this definition, concepts with high occurrence probability in the corpus are considered abstract concepts and therefore receive a low Information Content value. Information Content similarity measures require a corpus where use frequencies can be computed. In several domains corpus may not be available or be not representative enough to compute use frequencies. Further, IC measures omit neighborhoods like taxonomic similarity measures.

3.3.1 Resnik

Resnik [82] presents a similarity measure for nodes in a *is-a* a taxonomy. The likelihood of a node in a corpus is defined as $\text{p}(w) = \sum_{v \in \text{subclasses}(w)} \text{count}(v)$, where $\text{subclasses}(w)$ represents the set of nodes subsumed by w. In Figure 3.4, $\text{subclasses}(Europe)$ comprises *Italy*, *Russia* and *Turkey*. The similarity among two nodes is defined as:

$$\text{Sim}_{\text{Resnik}}(x,y) = \max_{w \in S(x,y)} -\log(\frac{\text{p}(w)}{N}),$$

where and N is the number of nodes in the corpus. The values returned by $\text{Sim}_{\text{Resnik}}$ are included in the interval $[0,\infty]$. Values close to 0 corresponds to nodes with general common ancestors, i.e., not very informative. High values corresponds to nodes with a very informative common ancestor, i.e., with a high information content value. The intuition behind is that, if a common ancestor is very informative, the two nodes share a lot of information. Similarity values are corpus dependent, i.e., two nodes can receive different similarity values depending on the used corpus to compute the information content values.

3.3.2 Redefinitions of Resnik's Measure

The similarity measure proposed by Resnik is based only on the Information Content of the common ancestors. The Information Content of the compared nodes is not considered. Lin [58] and Jiang & Conrath [42] refine the similarity measure proposed by Resnik by considering also the Information Content of the compared nodes.

Jiang & Conrath [42] define the similarity between two nodes x, y in a taxonomy as:

$$\text{Sim}_{\text{J\&C}}(x,y) = 1 + 2\text{Sim}_{\text{Resnik}}(x,y) - IC(x) - IC(y),$$

while Lin [58] presents the following definition:

$$\text{Sim}_{\text{J\&C}}(x,y) = \frac{2\text{Sim}_{\text{Resnik}}(x,y)}{IC(x)+IC(y)}.$$

Both similarity measures consider the same elements; together with the definition provided by Resnik, these similarity measures have been used in a wide range of applications related with NLP [83] and semantic similarity [79].

3.3.3 DiShIn

Previous Information Content based similarity measures do not take into account the multiple inheritance in a taxonomy and only consider the most informative common ancestor to compute the similarity between two nodes. To overcome this problem, Couto et al. [15] present the concept of *disjunctive common ancestors*. Two ancestors are considered disjunctive if the difference between the number of distinct paths from the compared nodes to it is different from that of any other more informative common ancestor. Formally, the set of disjunctive common ancestors is defined as follows:

$$DCA_{DiShIn}(x,y) = \{v : v \in CA(x,y) \wedge \forall w \in CA(x,y), PD(x,y,v) = PD(x,y,w) \rightarrow IC(v) > IC(w)\},$$

where CA represents the common ancestors and PD corresponds to the difference between the number of paths:

$$PD(x,y,v) = |\text{Paths}(x,v) - \text{Paths}(y,v)|.$$

In Figure 3.4, the LCA of *Italy* and *Russia* is *Europe*. From *Italy* to *World* there is only one path. However, from *Russia* to *World* there are two distinct paths. Thus, the disjunctive common ancestors are *Europe* and *World*.

The similarity between two nodes x, y is defined as the average of the Information Content values over the disjunctive common ancestors:

$$\text{Sim}_{\text{DiShIn}}(x,y) = \overline{\{IC(v) : v \in DCA_{DiShIn}(x,y)\}}.$$

3.4 Attribute Similarity Measures

Besides nodes and edges, a knowledge graph allows for defining attribute values for the described resources in form of literals. Literals can be number or portions of text. The value of these literals can also be relevant during

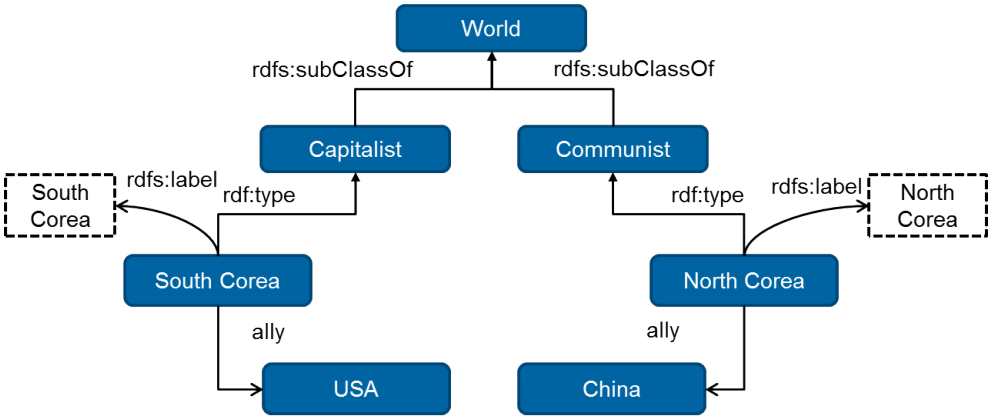


Figure 3.5: Portion of a knowledge graph that describe countries in the world according to their economic system and allies.

the computation of similarity values among knowledge graph resources. Nevertheless, similarity measures that only consider literals to determine similarity omit the information encoded in the hierarchy and the neighborhoods of the nodes. The knowledge graph in Figure 3.5 includes information about countries and their economic systems. According to their literals, *North Corea* and *South Corea* are similar. However, both taxonomic and path similarity measures will assign low similarity values. In this section an overview of text similarity measures is given.

Jaro-Winkler

There are several string similarity measures based on the edit distance or the difference in terms of characters among two strings. These measures are lexical and not semantic, i.e., the similarity value is based on the number of characters the two strings share and their position. The meaning of the string is not taken into account. One of these measures is the Jaro-Winkler distance [110], an extension of the Jaro distance [40].

The Jaro distance d_j among two strings s_1 and s_2 is defined as:

$$d_j(s_1,s_2) = \begin{cases} 0 & \text{if} \quad m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-1}{m}\right) & \text{otherwise} \end{cases}$$

where m is the number of matching characters and t the number of needed transpositions. Two characters c_1 and c_2 are a matching characters in strings s_1 and s_2 if they are the same $c_1 = c_2$ and their positions are not further than $\left\lfloor \frac{\max(|s_1|,|s_2|)}{2} - 1 \right\rfloor$. The number of transpositions t corresponds to the half of the number of matching characters between s_1 and s_2 that are in different sequence order. For example, the comparison of the strings *host* and *hots* returns four matching characters. Two of them, *s* and *t* are out of order. Thus, only one transposition is needed.

The Jaro-Winkler distance extends the Jaro distance giving more importance to the prefix of the string. Hence, coincident characters at the beginning of the string contributes strongly to the similarity value. The Jaro-Winkler distance d_w is among two strings s_1 and s_2 is formally defined as:

$$d_w(s_1,s_2) = d_j + lp(1-d_j),$$

where l is the length of the common prefix up to a maximum of four characters and p is a constant scaling factor to weight the importance of the prefix. To avoid similarity values larger than 1.0, p must not exceed 0.25.

Word2vec and Doc2Vec

In contrast to lexical string similarity measures, there are string similarity measures based on vector representations of the strings. The process of finding a vector representation for a string is called word embedding. Word embedding is based on the *Distributional Hypothesis*, which states that words in the same context share semantic meaning. There are two categories of approaches that work under this hypothesis: count-based methods (e.g. Latent Semantic Analysis) and predictive methods (e.g. Word2Vec [64]). Methods of both categories collect context-vectors from the corpus. Context-vectors are built from several statistical key values as the number of word co-occurrences in documents, paragraphs or sentences, their position in similar sentences, etc. The difference between count-based methods and predictive methods lays on the computation of the weights of the context-vectors. While weighting in count-based methods is performed based on human criteria, the weighting in predictive methods are set to optimally predict the contexts where the words tend to appear [7]. In the last years predictive methods have gained popularity and obtained better results than count-based methods. Therefore, this section focuses on predictive methods. Word2Vec [64] and Doc2Vec [53] are examples of predictive methods for word embedding. Word2Vec finds vector representations for words, while Doc2Vec is a generalization of Word2Vec for pieces of text of an arbitrary length like sentences, paragraphs or documents.

3.5 Combined Similarity Measures

The above described similarity measures compute the similarity values based on an individual aspect, either the taxonomy, the neighbors or the Information Content. Measures in this section combine two or more aspects when computing the similarity between nodes in a knowledge graph.

3.5.1 GBSS

Paul et al. [72] present GBSS, an efficient graph-based document similarity. Although it is originally designed to compare semantically annotated documents, it can be used to compare any semantically annotated resources. GBSS computes the similarity based on two aspects, the taxonomic or hierarchical similarity and the neighborhood or transversal similarity. Like path similarity measures, GBSS omit the relation type and understands all edges as they had the same semantics. Further, literals and informativeness are not taken into account.

The hierarchical similarity is generally defined as $Sim_{hier}(x,y) = 1 - d(x,y)$, where $d(x,y)$ is a hierarchical distance measure like D_{tax} [10] or D_{ps} [75]. The transversal similarity is based on the formula proposed by Nunes et al. [69] and is defined as follows:

$$Sim_{trans}(x,y) = \frac{\sum_{l=0}^{2L} \beta^l |\text{paths}^l(x,y)|}{\sum_{l=0}^{2L} \beta^l |\text{paths}^l(x,x)|},$$

where L is the length of the longest path to consider in the neighborhood and $\text{paths}^l(x,y)$ represents the paths of length l between x and y. Finally, the two aspects are combined to return a similarity value as follows:

$$GBSS(x,y) = Sim_{hier}(x,y) + Sim_{trans}(x,y)$$

3.6 Relation Discovery

A diversity of approaches has been proposed to solve the relation discovery task. Naive approaches are based on the hypothesis that states that similar entities should be related. These approaches use different similarity measures to discover relations between entities and their top-K most similar. The used similarity measure depends on the use case and considers knowledge encoded in the datatype properties or attributes to compute the similarity value. Examples of these kind of measures are Explicit Semantic Analysis (ESA) [26] and Doc2Vec, an implementation

of [53]. Both measures need a training corpus, and transform texts of an arbitrary length into vectors. Nevertheless, they do not consider simultaneously the knowledge encoded in other resource characteristics like GADES.

Palma et al. [71] and Flores et al. [24] present approaches for relation discovery in heterogeneous bipartite graphs. Palma et al. present semEP, a semantic-based graph partitioning approach that finds the minimal partition of a weighted bipartite graph with highest density. The density of a part in the partition is directly proportional to the average pairwise similarity among entities in the part. Hence, entities in the same part are highly similar and relations are discovered among them. However, semEP considers entities as isolated elements and does not consider their neighbors during the partitioning process. Flores et al. introduce esDSG [24]. an approach to find a portion of the graph that is highly dense and comprise highly similar entities. The density function is computed based on similarity values and the neighbors of the nodes in the knowledge graphs are omitted.

Researchers of the social network field study the structure of friendship induced graphs, and define the concept of ego-network as the set of entities that are at one-hop distance to a given entity, i.e., the neighborhood. Epasto et al. [20] report on high quality results in the friend suggestion task by analyzing the ego-networks of the induced knowledge graphs. In this case, the discovery of the relations is based purely on the neighborhood of the entities and attributes are not considered as relevant resource characteristic to determine the similarity or relatedness.

Redondo et al. [80] propose an approach to discover relations between video fragments based on visual information and background knowledge extracted from the Web of Data in form of semantic annotations. Similarly to semEP and esDSG [24, 71] entities or video fragments are considered as isolated elements in the knowledge graph, and the similarity is computed as the number of coincident annotations between two video fragments.

Sachan et al. [86] discover relations between authors in a co-author network extracted from dblp[1]. They consider the connections in the knowledge graph and some features of the authors and the papers like the keywords. However, the comparison of such features relies on the syntactic level, and the semantics is omitted.

Kastrin et al. [44] present an approach to discover relations among biomedical terms. They build a knowledge graph with such terms with the help of SemRep [84], a tool for recovering semantic propositions from the literature. In this case, not only the existence of the relation is important, but also the type of the relation. The proposed approach only considers the graph topology, discarding semantic knowledge encoded in attributes.

Nunes et al. [68] connect entities based on the number of co-occurrences in a text corpus and a distance measure proportional to the number of hops between them. Thus, only the graph topology is considered relevant to estimate the similarity and other resource characteristics like the relation types connecting the entities are omitted.

3.7 Evolution of Annotation Graphs

Škunca et al. [94] define a methodology to measure the evolution of an annotation graphs in terms of electronic annotations. The methodology focuses on the Gene Ontology (GO) and the UniProt dataset. Two *generations* of UniProt are the input of the methodology which relies on three measures of annotation quality: *i*) Reliability measures the fraction of electronic annotations that were confirmed in the most current generation by an experimental annotation. *ii*) Coverage measures the proportion of new experimental annotations that were predicted by some electronic annotation in the previous generation. *iii*) Specificity measures how informative are the electronically predicted annotations. At the same time, GO, the ontology used to describe the proteins, evolves and terms in the ontology are deleted, added or modified over time. Reliability is the only measure that considers the removing of annotations but it does not recognize if the deletion is due to the elimination of the term in GO or the original term evolved into another more specific GO term. Coverage also does not consider that some of the not predicted terms may not be available in the respective ontology version and, therefore the prediction was impossible.

Gross et al. [30] define a methodology to measure the quality of protein annotations. This methodology considers the annotation generation methods (provenance) and the evolution of the corresponding ontology. It is able to

[1] http://dblp.uni-trier.de/

indicate when the deletion of an annotation is due to a change in the ontology or not. However, it does not recognize if the addition of a new annotation is related to the addition of a term in the ontology. They use three indicators to measure the quality of an annotation: *i*) *Type provenance* is represented by the Evidence Codes in the case of Ensembl and SwissProt; *ii*) *Stability* can take the values stable and not stable; and *iii*) *Age* can be *novel*, *middle*, and *old*. To represent *type provenance* and *age*, Gross et al. [30] define also numerical measures and thresholds for the different categories of annotations.

Both methodologies are *annotation-oriented*, i.e., they analyze the evolution of each of the annotations. However, they omit the evolution of the whole annotation set and cannot differentiate when an annotation is replaced by a similar one or is just deleted.

3.8 Knowledge Graph Integration

Several approaches have been proposed to solve the problem of knowldge graph integration.

Knoblock et al. [48] propose *Karma*, a framework for the semi-automatic integration of structured data sources. Structured data sources lack usually of semantic descriptions. Karma facilitates the mapping of the entities of the properties and entities described in such data sources to RDF knowledge graphs. Afterwards, the created RDF knowledge graphs can be easier integrated thanks to the enriched semantic descriptions.

Schultz et al. [89] introduce the *Linked Data Integration Framework* (LDIF). The framework provides a set of pluggable modules to support the process of integrating RDF datasets. One of the key modules of LDIF is the entity resolution module, which has to detect when two entities in different RDF graphs represent the same real-world entity. The entity resolution module is implemented with *Silk*. *Silk* is an identity resolution framework based on heuristics that can be defined by the users. According to these heuristics, *Silk* looks for candidate entity descriptions in the different knowledge graphs that represent the same real-world entity.

Knap et al. [47] define *UnifiedViews*, an Extract-Transform-Load (ETL) framework for processing RDF data. *UnifiedViews* supports a wide range of processing tasks for RDF graphs, including the fusion or integration of RDF knowledge graphs. This task is performed by the *LD-FusionTool* [63], which relies on existing `owl:sameAs` links and similar graph structure to determine when to descriptions belong to the same real-world entity.

The above mentioned approaches require either domain knowledge and manual effort or rely only on syntactic properties of the knowledge graph, omitting the semantics encoded on them. The use of semantic similarity measures during the integration of knowledge graphs reduce allow to detect equivalent descriptions even when the descriptions are syntactically different, i.e., entities are described with non-matching URIs. Further, similarity-based approaches do not require to define heuristics or rules manually reducing the corresponding manual effort.

4 Semantic Similarity Measure Framework

In this chapter a semantic similarity measure framework containing four novel semantic similarity measures is described. OnSim and IC-OnSim compare ontology terms considering OWL semantics, while GADES enables the comparison of any two resources or entities in a knowledge graph. GARUM extends GADES by implementing a machine learning approach to determine the relevance of each resource characteristic when determining similarity values. Section 4.2 presents OnSim and IC-OnSim, while Section 4.3 and 4.4 describe GADES and GARUM, respectively. Each section includes a motivating example, the evaluation and a computational complexity analysis.

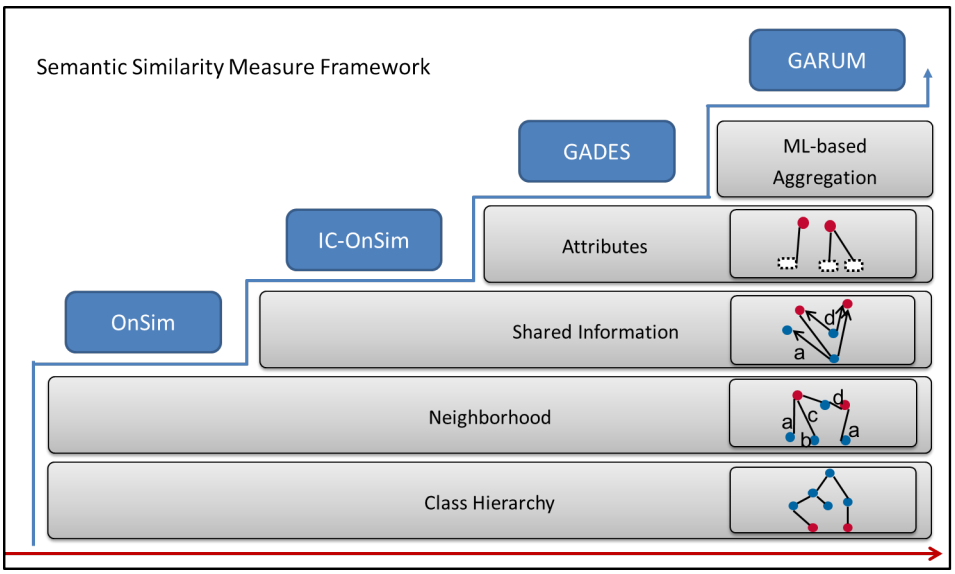


Figure 4.1: Semantic Similarity Measure Framework.

4.1 Introduction

Semantic Web technologies and Linked Data initiatives promote the publication of large volumes of data in the form of knowledge graphs. For example, knowledge graphs like DBpedia[1], Freebase or Yago[2], represent general domain concepts such as films, politicians, or sports, using RDF vocabularies. Additionally, domain specific communities like ife sciences and the financial domain, have also enthusiastically supported the collaborative development of diverse ontologies that can be included as part of the knowledge graphs to enhance the description of resources, e.g., the Gene Ontology (GO) [28], the Human Phenotype Ontology (HPO) [49], or the Financial Industry Business Ontology (FIBO) [3]. Knowledge graphs encode semantics that describe resources in terms of several *resource characteristics*, e.g., hierarchies, neighbors, attributes, and shared information. The impact of *resource characteristics* on the problem of determining relatedness between entities in a knowledge graph has been shown, and semantic similarity measures for knowledge graphs have been proposed, e.g., GBSS [72], HeteSim [92], and PathSim [97]. However, none of these measures considers all these *resource characteristics* at the same time. The

[1] http://dbpedia.org
[2] http://yago-knowledge.org
[3] https://www.w3.org/community/fibo/

	Hierarchy	Neighborhood	Shared Information	Attributes
OnSim	✓	✓	-	-
IC-OnSim	✓	✓	✓	-
GADES	✓	✓	✓	✓
GARUM	✓	✓	✓	✓

Table 4.1: Resource characteristics. Knowledge graph information considered by each similarity measure to estimate similarity values.

importance of precisely determining relatedness in data-driven tasks like clustering or ranking, and the increasing number of resources described in knowledge graphs, present the challenge of defining semantic similarity measures able to exploit these *resource characteristics*.

In this chapter, a framework including four similarity measures are presented. On one hand, OnSim and IC-OnSim are similarity measures for ontology terms able to consider OWL2 semantics and to make use of OWL reasoners to infer facts in the ontologies and obtain justifications of these facts. On the other hand, GADES and GARUM are similarity measures for knowledge graph resources based only on the explicitly represented knowledge, i.e., without making use of reasoning processes. GARUM makes use of supervised machine learning algorithm for regression purposes to determine similarity values. Thus, it needs to be trained beforehand. Because of that, GADES is the fastest similarity measure presented in this work, which makes it more suitable for near real-time applications. Further, unlike OnSim, IC-OnSim, and GARUM, GADES meets the properties of a metric. Thus, GADES can be consistently implemented in clustering algorithms. Table 4.1 shows the matching between *resource characteristics* and similarity measures. The chapter follows an evolutionary order starting with OnSim, the simplest measure in terms of considered resource characteristics, and ending with GARUM. The chapter also includes the evaluation of each measure and a study of their computational complexity.

4.2 Semantic Similarity Measures on Ontologies: OnSim and IC-OnSim

Ontology terms can be used as a controlled vocabulary to describe real world entities. Fuzzy search or data analysis tasks require of similarity measures to be able to deal with such descriptions, i.e., similarity measures for ontology terms. In this section two semantic similarity measures for ontology terms are described. First, OnSim is defined in Section 4.2.1. OnSim relies on the class hierarchy of the ontology and the neighbors of the ontology terms to compute similarity values. Next, IC-OnSim is described in Section 4.2.2. Unlike OnSim, IC-Onsim also computes the shared information among two ontology terms based on their use in a corpus. Additionally, each section includes a complexity analysis and an empirical evaluation of their performance in terms of time and accuracy.

4.2.1 OnSim

OnSim is the first similarity measure in the framework. OnSim considers two resource characteristics, the hierarchy and the neighborhoods. The measure is evaluated in the public benchmark CESSM [79][4][5]. OnSim is published in the proceedings of the International Conference on Data Integration in the Life Sciences [107].

[4] http://xldb.di.fc.ul.pt/tools/cessm/
[5] http://xldb.fc.ul.pt/biotools/cessm2014/

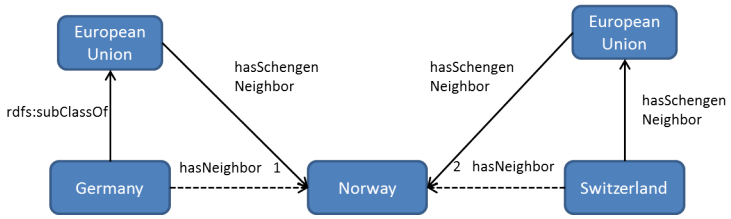


Figure 4.2: Portion of the neighborhood from Germany and Switzerland.

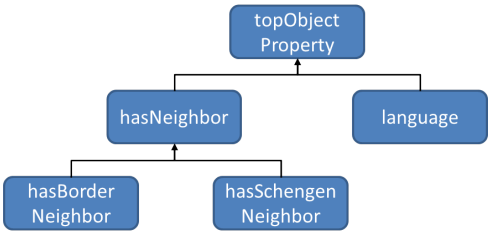


Figure 4.3: Hierarchy of Object Properties

Motivating Example

Figure 4.2 presents a portion of the neighborhoods of Germany and Switzerland according to the running example (see Figure 1.1). Solid arrows in Figure 4.2 represent stated object properties and dashed arrows represent inferred object properties. Countries can be related through different object properties. For example, in Figure 4.2 Germany is related to Norway through object properties `hasNeighbor` and `hasSchengenNeighbor`, which is a sub-property of `hasNeighbor` (Figure 4.3). Knowledge graphs can be described in OWL, which allows for representing logical axioms to describe the semantics of object properties, e.g., including logical axioms to express transitivity or symmetry. In this case, the object property `hasSchengenNeighbor` is defined as transitive, i.e., if two countries A and B are related through the `hasSchengenNeighbor` property, and B is also related through this property with a country C, then, A has to be related with C through the object property `hasSchengenNeighbor`.

Logical axioms enable reasoners to infer *facts* in the ontology. Apart from the inferred facts, reasoners also provide provenance information that explains why they are inferred. This provenance information is called justification. Figure 4.2 illustrates justifications of the inferred facts (Germany `hasNeighbor` Norway) and (Switzerland `hasNeighbor` Norway):

1. The first justification relies on: the *axiom of Instantiation of SubClassOf (`sc`) over `hasSchengenNeighbor`* and the *axiom of Instantiation of SubPropertyOf (`sp`) over `hasNeighbor`*. Observe the edges (Germany `sc` European Union) and (European Union `hasSchengenNeighbor` Norway) in Figure 4.2. Then, (Germany `hasSchengenNeighbor` Norway) can be inferred by transitivity of the object property `hasSchengenNeighbor` over `sc`. Finally, because `hasSchengenNeighbor` is a sub-property of `hasNeighbor`, the fact (Germany `hasNeighbor` Norway) is inferred.

2. The second inference is justified by the *axiom of Instantiation of Transitivity (`hasSchengenNeighbor`)*. Switzerland is related to the European Union through the object property `hasSchengenNeighbor`. The European Union is also related to Norway through the `hasSchengenNeighbor` object property. Given that the `hasSchengenNeighbor` is defined as transitive, the fact (Switzerland `hasSchengenNeighbor` Norway) is inferred. Finally, because `hasSchengenNeighbor` is a sub-property of `hasNeighbor`, the fact (Switzerland `hasNeighbor` Norway) is inferred.

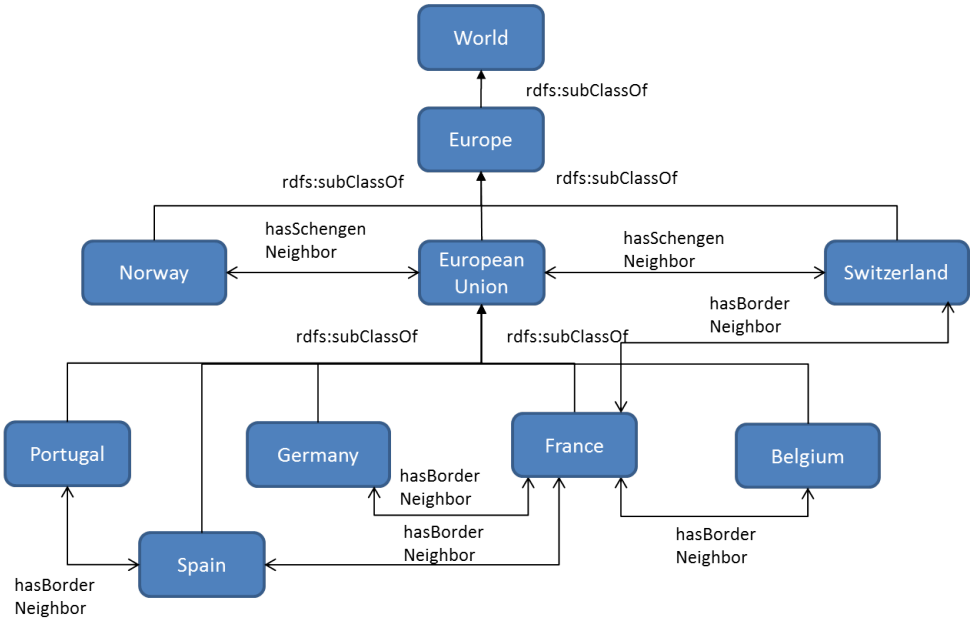


Figure 4.4: Portion of a knowledge graph that describe countries in the world according to their continent and neighbors.

Existing ontology-based similarities mainly rely on *taxonomic* hierarchies of classes, and are not aware of these differences. For example, D_{tax} [10] and D_{ps} [75] assign relatively high values of similarities to France and Switzerland in Figure 4.4, 0.4 and 0.25, respectively. Nevertheless, D_{tax} and D_{ps} ignore that both the neighborhoods of France and Switzerland, and the justifications of their inferred facts are different. Therefore, D_{tax} and D_{ps} values may overestimate the real value of relatedness among these countries. On the other hand, neighborhood based similarity measures like PathSim [97] or HeteSim [92], ignore the knowledge encoded in the hierarchy and may underestimate the similarity value. Thus, a combination of both resource characteristics is needed.

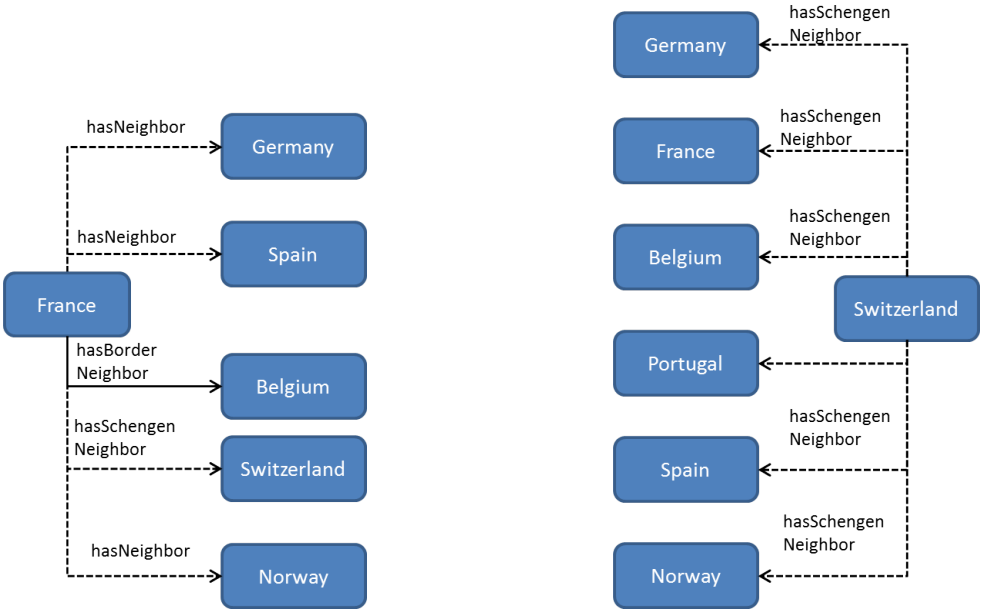
OnSim: An Ontology Similarity Measure

OnSim is a semantic similarity measure that computes relatedness between ontology terms. OnSim considers two resource characteristics: the hierarchy and the neighborhoods. The neighborhoods include the set of related terms, the label of each relation and, if the relation is inferred, the justification of the inference.

To illustrate the impact that considering additional knowledge may have on the computation of the similarity, consider the countries France and Switzerland. Figures 4.5(a) and 4.5(b) represent the neighborhoods of these terms. For the sake of clarity some object properties are substituted by their ancestors. The neighborhoods of these terms are different, as well as the justifications that support the inference of these facts. Nevertheless, taxonomic similarity measures ignore this information and may assign relatively high values of similarity to these two terms. Contrary, OnSim detects that these two terms are dissimilar regarding the neighbors, the object properties connecting to these neighbors, and their justifications, and assigns a lower similarity value., i.e., *OnSim(France, Switzerland)* is equal to *0.26* or *0.164*, depending on the used taxonomic similarity measure (D_{tax} or D_{ps}).

To represent neighborhoods and justifications, a set R_{a_i} is defined for each ontology term a_i. The set R_{a_i} represents the neighborhood of a_i, which consists of a set of facts. Facts in the neighborhood are modeled as quadruples $t = (a_i, a_j, r_{ij}, J_{ij})$, where r_{ij} is an object property such that the triple (a_i, r_{ij}, a_j) exists in the ontology, and J_{ij} is a set of the instantiations of the *antecedents* of the axioms used to infer the fact $(a_i\ r_{ij}\ a_j)$[6]. Thus, t_1=(France, Germany, `hasNeighbor`, {(`hasBorderNeighbor sp hasNeighbor`), (France `hasBorderNeighbor` Germany), Ax.4}) is the quadruple that represents that the countries France and Germany are related through the object property

[6] According to OWL2 semantics the inferred fact is: a_i subClassOf r_{ij} some a_j.



(a) Neighborhood of France (b) Neighborhood of Switzerland

Figure 4.5: Neighborhoods of countries France and Switzerland in running example.

`hasNeighbor` (Figure 4.5(b)). Further, t_1 states the justification of this inferred fact; in this case axiom Ax.4 is applied, and the instantiation of the antecedent of Ax.4 is (France `hasBorderNeighbor` Germany). A quadruple t is defined based on the OWL2 axioms applied in a given justification.

Definition 10. *Given two ontology terms a_i and a_j, and an object property r_{ij}. A fact in the neighborhood of a_i establishing that a_i and a_j are related through r_{ij}, i.e., $(a_i\ r_{ij}\ a_j)$, is represented as a quadruple $t = (a_i, a_j, r_{ij}, J_{ij})$, where J_{ij} is a set of the instantiations of the* antecedents *of the axioms used to infer the fact $(a_i\ r_{ij}\ a_j)$. Depending of the axioms used to inferred the fact $(a_i\ r_{ij}\ a_j)$, the quadruple t is inductively defined as follows:*

Ax.1 Axiom of Symmetry Relation r_{ij}:

$$\frac{(a_i\ r_{ij}\ a_j)}{(a_j\ r_{ij}\ a_i)} \implies \mathrm{t} = (\mathrm{a_i}, \mathrm{a_j}, \mathrm{r_{ij}}, \{(\mathrm{a_j\ r_{ij}\ a_i}), \mathrm{Ax.1}\})$$

Ax.2 Axiom of Instantiation of SubClassOf (sc) *over r_{ij}:*

$$\frac{(a_i\ sc\ a_z) \wedge (a_z\ r_{ij}\ a_j)}{(a_i\ r_{ij}\ a_j)} \implies \mathrm{t} = (\mathrm{a_i}, \mathrm{a_j}, \mathrm{r_{ij}}, \{(\mathrm{a_i\ sc\ a_z}), (\mathrm{a_z\ r_{ij}\ a_j}), \mathrm{Ax.2}\})$$

Ax.3 Axiom of Transitivity of SubClassOf (sc)*:*

$$\frac{(a_i\ sc\ a_z) \wedge (a_z\ sc\ a_j)}{(a_i\ sc\ a_j)} \implies \mathrm{t} = (\mathrm{a_i}, \mathrm{a_j}, \mathrm{sc}, \{(\mathrm{a_i\ sc\ a_z}), (\mathrm{a_z\ sc\ a_j}), \mathrm{Ax.3}\})$$

Ax.4 Axiom of Instantiation of SubPropertyOf (sp) *over r_{ij}:*

$$\frac{(r_z\ sp\ r_{ij}) \wedge (a_i\ r_z\ a_j)}{(a_i\ r_{ij}\ a_j)} \implies \mathrm{t} = (\mathrm{a_i}, \mathrm{a_j}, \mathrm{r_{ij}}, \{(\mathrm{r_z\ sp\ r_{ij}}), (\mathrm{a_i\ r_z\ a_j}), \mathrm{Ax.4}\})$$

Ax.5 Axiom of Transitivity of SubPropertyOf (sp)*:*

$$\frac{(a_i\ sp\ a_z) \wedge (a_z\ sp\ a_j)}{(a_i\ sp\ a_j)} \implies \mathrm{t} = (\mathrm{a_i}, \mathrm{a_j}, \mathrm{sp}, \{(\mathrm{a_i\ sp\ a_z}), (\mathrm{a_z\ sp\ a_j}), \mathrm{Ax.5}\})$$

Ax.6 Axiom of Transitivity Relation r_{ij}:

$$\frac{(a_i\ r_{ij}\ a_z) \wedge (a_z\ r_{ij}\ a_j)}{(a_i\ r_{ij}\ a_j)} \implies \mathrm{t} = (\mathrm{a_i}, \mathrm{a_j}, \mathrm{r_{ij}}, \{(\mathrm{a_i\ r_{ij}\ a_z}), (\mathrm{a_z\ r_{ij}\ a_j}), \mathrm{Ax.6}\})$$

Ax.7 Axiom of Transitivity of r_z over r_{ij}:

$$\frac{(a_i\ r_z\ a_z) \wedge (a_z\ r_{ij}\ a_j)}{(a_i\ r_{ij}\ a_j)} \implies \mathrm{t} = (\mathrm{a_i\, a_j}, \mathrm{r_{ij}}, \{(\mathrm{a_i\ \ r_z\ a_z}), (\mathrm{a_z\ r_{ij}\ a_j}), \mathrm{Ax.7}\})$$

Inductive Case: *If $t_z = (a_z, a_k, r_{zk}, J_{zk})$ is part of the neighborhood of a_z, $t_i = (a_i, a_j, r_{ij}, J_{ij})$ is in the neighborhood of a_i, and $(a_z\ r_{zk}\ a_k) \in J_{ij}$, then eliminate t_i from the neighborhood of a_i and add the quadruple $t = (a_i, a_j, r_{ij}, \overline{J}_{ij})$ to the neighborhood of a_i , where $\overline{J}_{ij} = (J_{ij} - \{(a_z\ r_{zk}\ a_k)\}) \cup J_{zk}$.*

Let us consider the countries France and Switzerland in Figures 4.5(a) and 4.5(b). The neighborhood of France represented by R_{France}, comprises 5 quadruples; the quadruples $t_{1.1}$ and $t_{1.2}$ describe the facts (France `hasNeighbor` Spain) and (France `hasNeighbor` Norway), respectively:

- $t_{1.1}$=(France, Spain,`hasNeighbor`, {(`hasBorderNeighbor sp hasNeighbor`), (France `hasBorderNeighbor` Spain), Ax.4}).
- $t_{1.2}$=(France, Norway,`hasNeighbor`, {`hasSchengenNeighbor sp hasNeighbor`), (France `sc` European Union), (European Union `hasSchengenNeighbor` Norway), Ax.2, Ax.4}.

Note that the quadruple $t_{1.2}$ includes the justification of the fact (France `hasSchengenNeighbor` Norway), which is supported by two axioms, and the inductive definition of a quadruple (Definition 10) is applied, i.e., the justification is as follows:

(France `sc` European Union) $\wedge$ (European Union `hasSchengenNeighbor` Norway)
$\Rightarrow$ <Ax.2, (A `sc` B) $\wedge$ (B r C) $\Rightarrow$ (A r C) >
(`hasSchengenNeighbor sp hasNeighbor`) $\wedge$ (France `hasSchengenNeighbor` Norway)
$\Rightarrow$ <Ax.4, (r_i `sp` r_j) $\wedge$ (B r_i C) $\Rightarrow$ (B r_j C) >
(France `hasNeighbor` Norway)

Similarly, $R_{Switzerland}$ describes the neighborhood of Switzerland and comprises 8 quadruples. The quadruple $t_{2.1}$ is described below and allows to infer the fact (Switzerland `hasSchengenNeighbor` Norway):

- $t_{2.1}$=(Switzerland, Norway, `hasSchengenNeighbor`, {(Switzerland `hasSchengenNeighbor` European Union), (European Union `hasSchengenNeighbor` Norway), Ax.6}).

Given two quadruples, $t_{1i} = (a_1, a_i, r_{1i}, J_{1i})$ and $t_{2j} = (a_2, a_j, r_{2j}, J_{2j})$, the similarity of two quadruples $Sim_{\mathrm{quad}}(t_{1i}, t_{2j})$ is defined as the product triangular norm, *TN*, that combines the taxonomic similarity of t_{1i} and t_{2j} with the similarity of the justification sets J_{1i} and J_{2j}, $\mathrm{Sim_{just}}(J_{1i}, J_{2j})$.

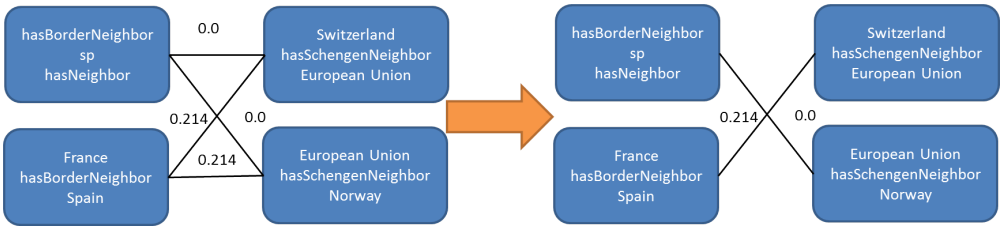


Figure 4.6: Comparison of the justifications of quadruples $t_{1.1}$ and $t_{2.1}$.

$$\mathbf{Sim_{quad}}(t_{1i}, t_{2j}) = \mathrm{Sim_{hier}}(a_i, a_j) \cdot \mathrm{Sim_{hier}}(r_{1i}, r_{2j}) \cdot \mathrm{Sim_{just}}(J_{1i}, J_{2j})$$

$\mathrm{Sim_{just}}$ aggregates similarity values among justifications items. An item it_i in a justification can be an axiom identifier, or an edge $(a_i\ r_{ij}\ a_j)$ that denotes the instantiation of one of the antecedents of the axiom. For example, the justification of the quadruple $t_{1.1}$=(France, Spain,`hasrNeighbor`, {(`hasBorderNeighbor sp hasNeighbor`), (France `hasBorderNeighbor` Spain), Ax.4}) is a set that comprises three items; two items are facts or edges (`hasBorderNeighbor sp hasNeighbor`) and (France `hasBorderNeighbor` Spain), Ax.4}), and the other item is the identifier of the applied axiom, i.e., Ax.4. Then, the similarity of two justification items is defined as follows:

$$\mathbf{Sim_{item}}(it_i, it_j) = \begin{cases} \mathrm{Sim_{edge}}(it_i, it_j) & \text{if} \quad \mathrm{edge}(it_i) \wedge \mathrm{edge}(it_j) \\ 1 & \text{if} \quad \mathrm{axiom}(it_i) = \mathrm{axiom}(it_j) \\ 0 & \text{otherwise} \end{cases},$$

i.e. if both items are edges $\mathrm{Sim_{edge}}$ is returned. In case both items correspond to the same axiom identifier $\mathrm{Sim_{item}}$ returns 1. Otherwise, either different types of items are compared and 0 is returned. The similarity of two edges $(a_i\ r_{ij}\ a_j) \in it_i$ and $(a_x\ r_{xy}\ a_y) \in it_j$, named $\mathrm{Sim_{edge}}$, is defined as the product triangular norm that combines three taxonomic similarities: $\mathrm{Sim_{hier}}(a_i, a_x)$, $\mathrm{Sim_{hier}}(r_{ij}, r_{xy})$, and $\mathrm{Sim_{hier}}(a_j, a_y)$:

$$\mathbf{Sim_{edge}}((a_i\ r_{ij}\ a_j), (a_x\ r_{xy}\ a_y)) = \mathrm{Sim_{hier}}(a_i, a_j) \cdot \mathrm{Sim_{hier}}(r_{ij}, r_{xy}) \cdot \mathrm{Sim_{hier}}(a_j, a_y)$$

In the running example, if the taxonomic similarity $\mathrm{Sim_{hier}}$ is D_{tax} [10], the similarity of the justification items it_1=(France `hasBorderNeighbor` Spain) and it_2=(Switzerland `hasSchengenNeighbor` European Union) is 0.214:

- $\mathrm{Sim_{hier}}$(France, Switzerland) is 0.4;
- $\mathrm{Sim_{hier}}$(`hasSchengenNeighbor`,`hasNeighbor`) is 0.67;
- $\mathrm{Sim_{hier}}$(Spain, European Union) is 0.8;
- $\mathrm{Sim_{item}}(e_1, e_2)$=$0.4 \times 0.67 \times 0.8 = 0.214$.

Two justifications J_{1i} and J_{2j} are compared based on the similarity values of their items. Formally, the similarity of two justifications is computed from a bipartite graph that corresponds to the 1-1 *maximum weight bipartite matching* of the edges in the Cartesian product of $J_{1i} \times J_{2j}$. Figure 4.6 shows the comparison of the justifications of quadruples $t_{1.1}$ and $t_{2.1}$. The figure is divided into two bipartite graphs: a) Bipartite graph from the pair-wise

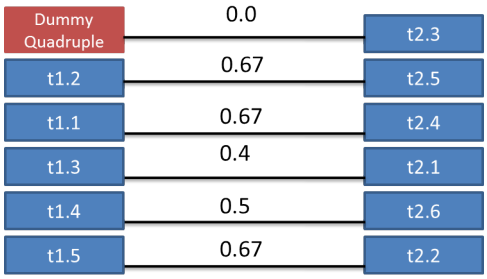


Figure 4.7: Comparison of R_{France} and $R_{Switzerland}$: 1-1 maximum weight bipartite matching produced by the Hungarian Algorithm [50].

comparison of the justifications; b) 1-1 MWBM produced by the Hungarian Algorithm [50]. Axiom identifiers are omitted for legibility. The justification sets $t_{1.1}$ and $t_{2.1}$ are described as follows:

$$t_{1.1} = (\text{France}, \text{Spain}, \texttt{hasNeighbor}, \{(\texttt{hasBorderNeighbor sp hasNeighbor}), (\text{France } \texttt{hasBorderNeighbor} \text{ Spain}), Ax.4\})$$

$$t_{2.1} = (\text{Switzerland}, \text{Norway}, \texttt{hasSchengenNeighbor}, \{(\text{Switzerland } \texttt{hasSchengenNeighbor} \text{ European Union}), (\text{European Union } \texttt{hasSchengenNeighbor} \text{ Norway}), Ax.6\})$$

The computation of the *1-1 maximum weight bipartite matching* (MWBM) from a bipartite graph is solved using the Hungarian Algorithm [50]. Each edge is weighted with the corresponding item similarity value. Once the 1-1 maximum weight bipartite matching *MWBM* of $J_{1i} \times J_{2j}$ is computed, the similarity of these justifications is calculated as follows.

$$\mathbf{Sim_{just}}(J_{1i}, J_{2j}) = \frac{\sum_{(e_i,e_j) \in MWBM(J_{1i},J_{2j})} \text{Sim}_{\text{item}}(e_i, e_j)}{Max(|J_{1i}|, |J_{2j}|)}$$

Particularly, the Sim_{just} values for the 1-1 MWBM of quadruples $t_{1.1}$ and $t_{2.1}$ in Figure 4.6 is 0.06.

The neighborhood similarity $\text{Sim}_{\text{neigh}}$ is computed building a bipartite graph *GOS*=($R_1 \cup R_2$, *EOS*) where *EOS* is equivalent to the cartesian product of the neighborhoods $R_1 \times R_2$. Edges in *EOS* are weighted with the Sim_{quad} similarity values. Then, using the Hungarian Algorithm, a 1-1 maximum weighted perfect matching is computed in *GOS*. The result is a subset of edges $MEOS \subseteq EOS$ that corresponds to the 1-1 maximum weight bipartite matching of *GOS*. $\text{Sim}_{\text{neigh}}$ is defined as follows:

$$\mathbf{Sim_{neigh}}(a_1, a_2) = \frac{\sum_{(t_{1i},t_{2j}) \in MEOS} Sim_{\text{quad}}(t_{1i}, t_{2j})}{Max(|R_1|, |R_2|)},$$

where

- R_1 and R_2 are the sets of quadruples associated with a_1 and a_2, respectively;
- *MEOS* corresponds to the 1-1 maximum weight bipartite matching of the quadruples in the Cartesian product of R_1 and R_2 annotated with the similarity $Sim(t_{1i}, t_{2j})$;
- quadruples $t_{1i} = (a_1, a_i, r_{1i}, J_{1i})$ and $t_{2j} = (a_2, a_j, r_{2j}, J_{2j})$ belong to *MEOS*; and

- $Sim_{\text{quad}}(t_{1i}, t_{2j})$ is defined as a triangular norm TN[7] and is used to combine similarity values of the justifications in r_{1i}, r_{2j} with the taxonomic similarity of t_{1i} and t_{2j}.

Figure 4.7 presents the 1-1 MWBM found by the Hungarian Algorithm [50] for R_{France} and $R_{Switzerland}$. Observe that two dummy nodes are added to ensure that the sum of the similarity values is maximized. $\text{Sim}_{\text{neigh}}$ is computed on top of this 1-1 maximum weight bipartite matching and returns a value of 0.488.

Finally, the similarity $OnSim(a_1, a_2)$ is defined as a triangular norm TN that combines the taxonomic similarity Sim_{hier} and the neighborhood similarity $\text{Sim}_{\text{neigh}}$:

$$\textbf{OnSim}(a_1, a_2) = TN\Big(\text{Sim}_{\text{hier}}(a_1, a_2), \text{Sim}_{\text{neigh}}(a_1, a_2)\Big)$$

Supposing that Sim_{hier} corresponds to D_{ps}, the similarity value returned by OnSim is 0.25×0.656 =*0.164*, which is lower than the values of D_{tax} and D_{ps} reported in Subsection 4.2.1 (0.4 and 0.25 respectively).

4.2.2 IC-OnSim

In Section 4.1 four resource characteristics are identified as relevant to estimate similarity values between entities in a knowledge graph. OnSim considers two of these aspects: the hierarchy and the neighborhood. Nevertheless, the shared information is omitted. IC-OnSim [104] extends OnSim by taking also into account this resource characteristic. The evaluation shows that IC-OnSim outperforms OnSim in both CESSM benchmarks (Section 4.2.4), which means that considering the shared information has a positive impact when computing similarity values. IC-OnSim is published in the proceedings of the IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology [104].

Motivating Example

Let suppose a corpus of economy articles annotated with the countries involved in the article. Figure 4.8 presents two news articles annotated with the countries about the articles talk about. According to the taxonomy presented in Figure 4.9, taxonomic similarity measures as D_{tax} [10] and D_{ps} [75] return high similarity values for the comparisons among pairs Article1-Article2 and Article3-Article4. Portugal and Spain are equally similar as France and Germany when considering the hierarchical properties. Further, Portugal and Spain do not share any neighbor and Germany and France share several ones (Belgium, Luxembourg and Switzerland). Therefore, considering the neighborhoods does not allow to untie the results obtained with the taxonomic measures and OnSim returns 0.75 and 0.59 for both comparisons respectively. However, because of the importance of France or Germany in the world economy, it is much more probable to find articles about them in the corpus.

OnSim bases its similarity estimations among terms on hierarchical and neighborhood similarity and omits the informativeness of such terms. As explained in Section 3.3, Information Content based similarity measures pay also attention to the informativeness, based on the frequency of use of a certain term in a corpus to estimate the abstraction or specificity of the terms. Thus, coincidences of specific terms are more relevant than coincidences of general terms. Resnik [82] measures relatedness among ontology terms as the Information Content of their *Most Informative Common Ancestor* (MICA). The intuition behind, is that terms with an informative common ancestor share a lot of information. The MICA of Portugal and Spain is *1986 Members*, while the MICA of Germany and France is *Founder Members*. Given that the corpus contains news articles about economy and the economic weight of the founder members of the EU is heavier than the 1986 members, the amount of news articles talking about some of founder members will be higher. According to this setting, the term *Founder Members* is considered to be more abstract (less specific or informative) than the term *1986 Members* and so is reflected by the similarity

[7] For this ontology the *Product TN*

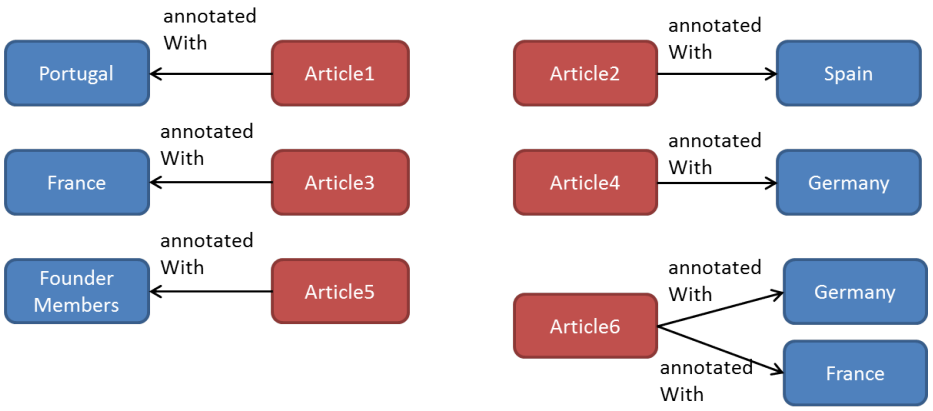


Figure 4.8: Portion of an annotation graph of news articles annotated with the countries about they talk.

measure defined by Resnik [82][8], which will return a lower value for the comparison between France and Germany than for Portugal and Spain. Considering Information Content in conjunction with the semantics encoded in the neighborhoods and taxonomies may enhance the accuracy of similarity values and, in consequence, of aggregated measures like the 1-1 maximum weighted matching.

IC-OnSim: An Ontology-based Information-aware Similarity Measure

IC-OnSim is a semantic similarity measure that computes relatedness between ontology terms. IC-OnSim extends OnSim by considering also the shared information when computing the similarity among two ontology terms.

To illustrate the impact that considering additional knowledge may have on the computation of the similarity, suppose two news articles that are annotated with the countries France and Germany. These countries have similar neighborhoods, returning $Sim_{neigh}(\text{France}, \text{Germany}) = 0.78$. D_{tax} returns 0.75, which means that the LCA is relatively deep and close to the terms. Unlike to Resnik's similarity measure, *OnSim* ignores that these two terms share few information and assigns relatively high values of similarity to these two terms. IC-OnSim extends *OnSim* by combining values of *OnSim* with the Resnik's similarity measure during the computation of the similarity values of two ontology terms. Thus, IC-OnSim is able to consider the *shared information* by the two ontology terms during the computation of the similarity. In order to measure the Information Content of a term is necessary to have a corpus available where the compared terms are used. The information shared by two ontology terms is measured as the Information Content of their *Most Informative Common Ancestor* (MICA), i.e.:

$$\mathbf{Sim_{shared}}(a_1, a_2) = \max_{w \in CA(a_1,a_2)} -\log(\frac{freq(w)}{N}),$$

where $CA(a_1, a_2)$ returns the common ancestors of a_1 and a_2, N is the number of annotated entities in the corpus and $freq(w)$ represents the frequency of an ontology term in the corpus. The *frequency* (*freq*(x)) of an ontology term x is calculated using the expression proposed by Mazandu et al. [60].

$$[h]\mathbf{freq}(x) = \begin{cases} A(x) & \text{if } x \text{ is a leaf} \\ A(x) + \sum_{z \in D(x)} A(z) & \text{otherwise,} \end{cases}$$

$A(x)$ ...Number of entities annotated with the ontology term x.
$D(x)$ ...Set of ontology terms that have x as indirect or direct ancestor.

[8] To be consistent with the rest of measures Sim_{Resnik} is normalized to return values in $[0, 1]$

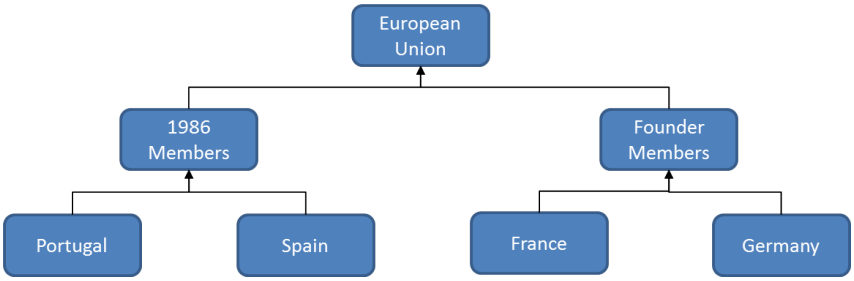


Figure 4.9: Portion of a class hierarchy of the running example in Figure 1.1

Comparison	Sim'_{Resnik}	D_{tax}	Sim_{neigh}	IC-OnSim
France-Germany	0.176	0.75	0.78	0.103

Table 4.2: Similarity values for the comparison between countries France and Germany.

Following this formula, to calculate the IC of *Founder Members* in Figure 4.9, the number of news articles annotated with *Founder Members* and the number of news articles annotated with some of its descendants (i.e., France, Germany, and so on) are considered.

Given a triangular norm TN, IC-OnSim(a_1, a_2) is defined as follows:

$$\textbf{IC-OnSim}(a_1,a_2) = TN\Big(TN(Sim_{hier}(a_1,a_2), Sim_{neigh}(a_1,a_2)), Sim_{shared}(a_1,a_2),\Big)$$

Table 4.2 contains the similarity results of each resource characteristic and their aggregation means the product triangular norm. Observe that, even when the hierarchical and the neighborhood are high, the information shared by the two terms according to the Resnik's measure is rather low.

4.2.3 Theoretical Properties

One of the research questions of this thesis is about how semantic similarity measures can scale up in large datasets and wether they can be used for real-time applications. This section includes a study about the computational complexity of OnSim and IC-OnSim. This computational complexity study is accompanied by an empirical study on performance in Section 4.3.4.

Time Complexity

In this section the computational complexity of OnSim and IC-OnSim is described. Given that IC-OnSim extends OnSim, the computational complexity of OnSim is studied first.

OnSim OnSim considers the hierarchy and the neighborhood as relevant resource characteristics to estimate the similarity. Further, OnSim requires the execution of an OWL reasoner to infer implicit edges and their justifications. This inference process happens before the similarity values are computed. Thus, OnSim consists of two phases: *i*) the computation of the neighborhood and justifications of all the inferences in the neighborhood for each compared ontology term and *ii*) computation of the ontology term similarity values based on the previously calculated neighborhoods, justifications and the hierarchy.

During the first phase the neighborhood, the compound of stated and inferred facts, and the justifications of the inferred facts are computed. The execution of an OWL reasoner is necessary for the computation of the inferred facts in the neighborhood and their justifications. Before computing the neighborhoods it is required to classify the ontology. Considering OWL2 semantics, the ontology classification has a worst case complexity of

2NEXP-Time [45]. Let $G = (V, E, L)$ be the knowledge graph representing the ontology terms to be compared. Let $V' \subseteq V$ be the set of ontology terms to be pairwise compared. Before performing the pairwise comparison, the neighborhood of each term is computed. Given an ontology term $v_1 \in V'$, the neighborhood is defined as a set of facts $R(v_1) = \{t = (v_1, v_j, r_{1j}, J_{ij}) \mid (v_1, r_{1j}, v_j) \in E\}$, where $v_j \in V$ is an ontology term, $r_{1j} \in L$ is an edge label and J_{ij} represents the set of axioms that justify the inference of (v_1, r_{1j}, v_j). The inference of implicit edges requires checking the entailment of axioms in the ontology. The reasoner is asked about the entailment of axioms of type $t_1 \sqsubseteq \exists r.t_2$, where t_1 and t_2 are ontology terms and r represents an edge label. Checking the entailment of this kind of axioms has a complexity of NEXP-Time [37]. Thus, computing the neighborhoods of the compared ontology terms has a worst-case complexity of:

$$O(\overbrace{2NEXP}^{\text{ontology classification}} + \overbrace{|V'| * |V| * |L| * NEXP}^{\text{checking axiom entailment}})$$

Limiting the expressiveness to the OWL EL profile allows to reduce the complexity of the initialization phase. The classification of OWL EL ontologies and the checking of entailments can be performed in polynomial time. Further, Zhou et al. [112] described a method to reduce also the complexity of the computation of the justifications. Thus, the computational complexity of the initialization phase when considering OWL EL semantics is reduced to polynomial time and it is described as $O(|V|^a + |V|^2|E||V|^b)$, where a and b are constants.

Once the neighborhoods are computed, the ontology terms V' can be pairwise compared. Thus, in total $\frac{|V'|(|V'|+1)}{2}$ comparisons are performed. For the sake of clarity in this section the computational complexity of one comparison is described. OnSim considers the hierarchical similarity Sim_{hier} and the neighborhood similarity $\text{Sim}_{\text{neigh}}$ to estimate the similarity between two ontology terms. Then, the worst-case computational complexity is described as follows:

$$O(\text{OnSim}) = O(\text{Sim}_{\text{hier}}) + O(\text{Sim}_{\text{neigh}})$$

Hierarchical similarity measures like D_{tax} and D_{ps} are based on the distance of the terms to their LCA. Let d be the maximum depth of the hierarchy. In the worst case, the terms are at depth d and V is their set of common ancestors. Then, the computational complexity of Sim_{hier} for two ontology terms is $O(|V|d)$. Han [33] showed that it is possible to compute the LCA of all the nodes in V with a worst-case complexity of $O(|V|^{2.575})$.

Algorithm 1 describes the function $\text{Sim}_{\text{neigh}}$. The algorithm consists of two phases. First, the function Sim_{quad} is called for each pair of neighbors. In second place the Hungarian Algorithm, whose computational complexity is $O(|V|^3)$, is called to compute the 1-1 maximum weighted matching among the neighbors of the compared terms. In the worst case the compared terms have so many neighbors as the ontology terms has, i.e., $N = |V|$. Thus, the computational complexity of $\text{Sim}_{\text{neigh}}$ is described as follows:

$$O(\text{Sim}_{\text{neigh}}) = O(|V|^2)O(\text{Sim}_{\text{quad}}) + O(\text{Hungarian Algorithm}),$$

Algorithm 2 describes the function Sim_{quad}. Considering the complexity of the assignments constant, the computational complexity of Sim_{quad} is described as:

$$O(\text{Sim}_{\text{quad}}) = O(\text{Sim}_{\text{hier}}) + O(\text{Sim}_{\text{just}})$$

Algorithm 1 $\text{Sim}_{\text{neigh}}$

procedure $\text{SIM}_{\text{NEIGH}}(a_1, a_2)$
  $N_1 \leftarrow \text{Neighbors}(a_1)$
  $N_2 \leftarrow \text{Neighbors}(a_2)$ ▷ For the sake of clarity, it is assumed that $|N_1| = |N_2| = |N|$
  for $i = 0; i < N; i = i + 1$ **do**
    for $j = i; j < N; j = j + 1$ **do**
      $sim[i, j] = \text{Sim}_{\text{quad}}(n_i, n_j)$
    end for
  end for
  return MatchingHungarian(N_1, N_2, sim)
end procedure

Algorithm 2 Sim_{quad}

procedure $\text{SIM}_{\text{QUAD}}(n_1, n_2)$
  $rel_1 \leftarrow EdgeLabel(n_1)$
  $rel_2 \leftarrow EdgeLabel(n_2)$
  $target_1 \leftarrow Target(n_1)$
  $target_2 \leftarrow Target(n_2)$
  $just_1 \leftarrow Justification(n_1)$
  $just_2 \leftarrow Justification(n_2)$
  return $\text{Sim}_{\text{hier}}(rel_1, rel_2)\text{Sim}_{\text{hier}}(target_1, target_2)Sim_{just}(just_1, just_2)$
end procedure

Algorithm 3 corresponds to the function Sim_{just}. Like Algorithm 1, the algorithm can be split into two phases: the pairwise computation of the similarity between justification items and the execution of the Hungarian Algorithm. In the worst case, the justifications contains all the edges represented in the ontology. Thus, the computational complexity of Sim_{just} is described as:

$$O(\text{Sim}_{\text{just}}) = O(|E|^2)O(\text{Sim}_{\text{item}}) + O(E^3),$$

where $O(\text{Sim}_{\text{item}}) = O(\text{Sim}_{\text{hier}}) = O(|V|d)$

Algorithm 3 Sim_{just}

procedure $Sim_{just}(j_1, j_2)$
  $It_1 \leftarrow Items(j_1)$
  $It_2 \leftarrow Items(j_2)$
  for $i = 0; i < N; i = i + 1$ **do**
    for $j = i; j < N; j = j + 1$ **do**
      $sim[i, j] = Sim_{item}(it_i, it_j)$ ▷ Sim_{item} has the same complexity as Sim_{hier} $O(|V|d)$
    end for
  end for
  return MatchingHungarian(It_1, It_2, sim)
end procedure

Table 4.3 contains the computational complexity of each of the functions computed by OnSim. Thus, considering that the ontology is classified and the neighborhoods are computed, the computational complexity of OnSim is $O(|V|^3|E|^2d) + O(|V|^2E^3)$.

Function	Complexity
Initialization	$O(2NEXP + \|V'\| * \|V\| * \|L\| * NEXP)$
Sim_{item}	$O(\|V\|d)$
Sim_{just}	$O(\|E\|^2\|V\|d) + O(E^3)$
Sim_{quad}	$O(\|V\|d) + O(\|E\|^2\|V\|d) + O(E^3)$
$\text{Sim}_{\text{neigh}}$	$O(\|V\|^3\|E\|^2 d) + O(\|V\|^2 E^3)$
OnSim	$O(\|V\|^3\|E\|^2 d) + O(\|V\|^2 E^3)$

Table 4.3: Computational complexity of each function computed by OnSim.

Function	Complexity
Initialization	$O(2NEXP + \|V'\| * \|V\| * \|L\| * NEXP + \|V\| * d + \|V\|^2 * \|A\|)$
IC-OnSim	$O(\|V\|^3\|E\|^2 d) + O(\|V\|^2\|E\|^3)$

Table 4.4: Computational complexity of each function computed by IC-OnSim.

IC-OnSim IC-OnSim extends OnSim computing additionally the Information Content of the Lowest Common Ancestor of the compared terms. Hence, apart from classifying the ontology and computing the neighborhoods during the first phase, IC-OnSim computes the Information Content of each term in the ontology. The Information Content is computed based on a corpus of annotated entities A. To compute the Information Content of each term, the frequency of each term in the corpus is calculated. Given that the terms are ordered in the hierarchy of the ontology, the occurrence of a term has to be considered also as occurrence of each one of its ancestors. Computing the ancestors of an ontology term has a computational complexity of $O(d)$, where d is the maximum depth of the hierarchy. The worst case is represented by a term that has V as ancestor set and that is part of the annotation set of all the entities A. The number of frequency updates to be performed is then $|V| * |A|$. Then, the worst-case complexity of computing the Information Content of one term is $O(d + |V| * |A|)$. As this process has to be performed for each term in V, the worst-case complexity is $O(|V| * d + |V|^2 * |A|)$. Thus, the computational complexity of the first phase is:

$$O(\overbrace{2NEXP}^{\text{ontology classification}} + \overbrace{|V'| * |V| * |L| * NEXP}^{\text{checking axiom entailment}} + \overbrace{|V| * d + |V|^2 * |A|}^{\text{Information Content}})$$

The computational complexity of the second phase of IC-OnSim is described as follows:

$$O(\text{IC-OnSim}) = O(\text{Sim}_{\text{hier}}) + O(\text{Sim}_{\text{shared}}) + O(\text{Sim}_{\text{neigh}})$$

Note that the only difference with respect to $O(\text{OnSim})$ is the insertion of $O(\text{Sim}_{\text{shared}})$, whose worst case computational complexity is $O(|V|d)$. Table 4.4 summarizes the computational complexity of IC-OnSim.

4.2.4 Empirical Properties

In this section the empirical properties of OnSim and IC-OnSim are presented. First, the efficiency of both similarity measures is shown in two versions of the CESSM benchmark [79]. Second, the performance of both measures in terms of time is empirically analyzed.

Evaluation on Ontology-based Annotated Datasets

The goal of the study is to evaluate the impact of OnSim and IC-OnSim on existing annotation-based similarity measures. The research hypothesis states that, because OnSim and IC-OnSim consider the neighborhood of two ontology terms and, in the case of IC-OnSim, the shared information, the annotation-based similarity values of entities annotated with such terms are more accurate.

Design of the experiment: The empirical study is conducted on the collections of proteins published at the CESSM portals of 2008[9] and 2014[10] using HermiT 1.3.8 [29] as OWL reasoner to infer neighborhood facts. CESSM portals are online tools for the automated evaluation of semantic similarity measures over proteins annotated with GO terms. Annotations are extracted from UniProt-GOA, and are separated into the GO hierarchies of Biological Process (BP), Molecular Function (MF), and Cellular Component (CC). Both version of CESSM include values of gold standard as well as state-of-the-art similarity measures. The quality or accuracy of each similarity measure is determined as the Pearson's correlation coefficient of each measure with respect to the three gold standards: *ECC* similarity [17], *Pfam* similarity [79], and the Sequence Similarity *SeqSim* [78]:

- *ECC* or *Enzyme Comparison Class* measures the similarity among two proteins according to their EC number, which is a numerical classification scheme for enzymes based on the chemical reactions they catalyze [66]. ECC returns values among 0 and 4 that corresponds to the number of digits that the compared proteins share. Let two proteins $p1$ and $p2$ have EC number 1.3.4.10 and 1.3.4.8. The ECC similarity among $p1$ and $p2$ would be 3, since they share the three first digits.
- *Pfam* returns values between 0 and 1 and is computed as the Jaccard coefficient among the Pfam families shared by the compared proteins. Pfam is a database of protein families [22]. Proteins are composed of one or more functional regions in their sequences. These regions are also called domains. Thus, the occurrence of a certain domain in a protein sequence can provide insights about their function. Pfam families contains proteins sharing domains and, therefore, functionality.
- *SeqSim* or Sequence similarity returns values among 0 and 1 is computed using RRBS [78]. RRBS measures the similarity between two protein sequences according to their BLAST scores. BLAST is the abbreviation of Basic Local Alignment Search Tool, an algorithm for comparing biological sequences.

Dataset description: The CESSM 2008 collection contains 13,430 pairs of proteins from UniProt with 1,039 distinct proteins, while the CESSM 2014 collection comprises 22,302 pairs with 1,559 distinct proteins. The annotation dataset of CESSM 2008 contains 1,908 distinct GO terms and the dataset of 2014 includes 3,909 GO terms. Annotations are restricted to GO Biological Process (BP) terms, the richer branch of GO in terms of axioms. Information Content values are calculated considering only the annotations of the proteins present in CESSM 2008 and 2014 as corpus, respectively. The complete UniProt dataset is not considered to be consistent with the rest of IC-based similarity measures in the benchmark. GO includes in its 2008 version four object properties and a class hierarchy with a maximum depth of 15 levels. One of the object properties, (*part of*) is transitive, i.e., the Axiom of Transitivity (see Ax.6 in Section 4.2.1) can be applied. The depth of the class hierarchy of 2014 version increases until 17 levels and the number of object properties increases to ten properties. Three of these properties are transitive, i.e., the Axiom of Transitivity (Ax.6 in Section 4.2.1) is also used.

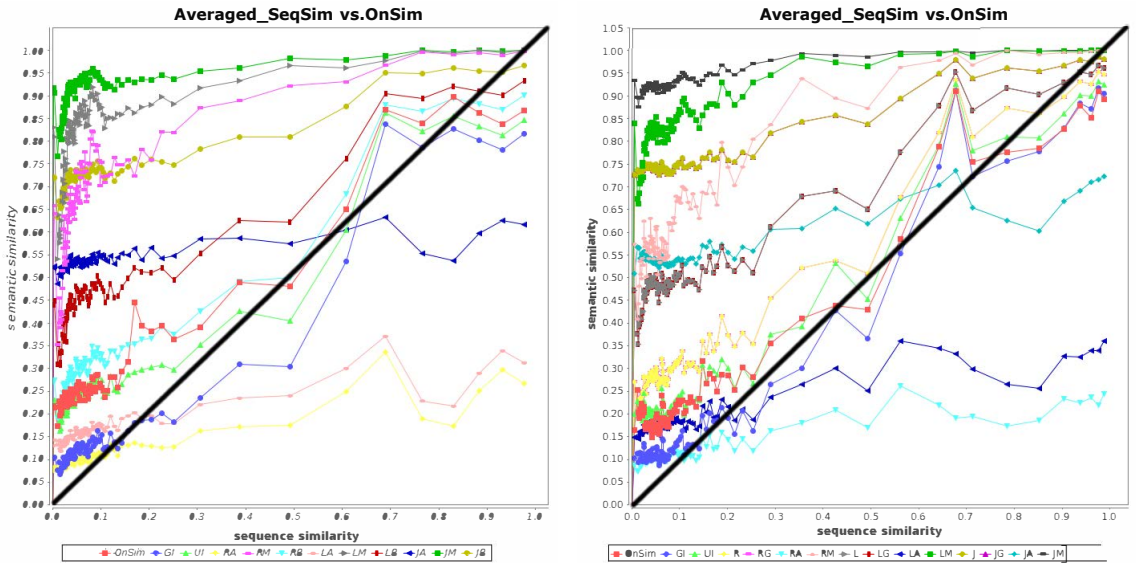
Eleven semantic similarity measures are compared in CESSM 2008 and fourteen in CESSM 2014. These similarity measures includes extensions of Resnik's (R) [82], Lin's (L) [58], and Jiang and Conrath's (J) [42] measures to consider GO annotations of the compared proteins and the Information Content (IC) of these annotations. These

[9] http://xldb.di.fc.ul.pt/tools/cessm/
[10] http://xldb.di.fc.ul.pt/biotools/cessm2014/

Resource Characteristic	Similarity Measure
Shared Information	GI [77]
	RA, RM, RB [82]
	JA, JM, JB [42]
	LA, LM, LB [58]
Hierarchy	AnnSim [70]
None	UI [77]

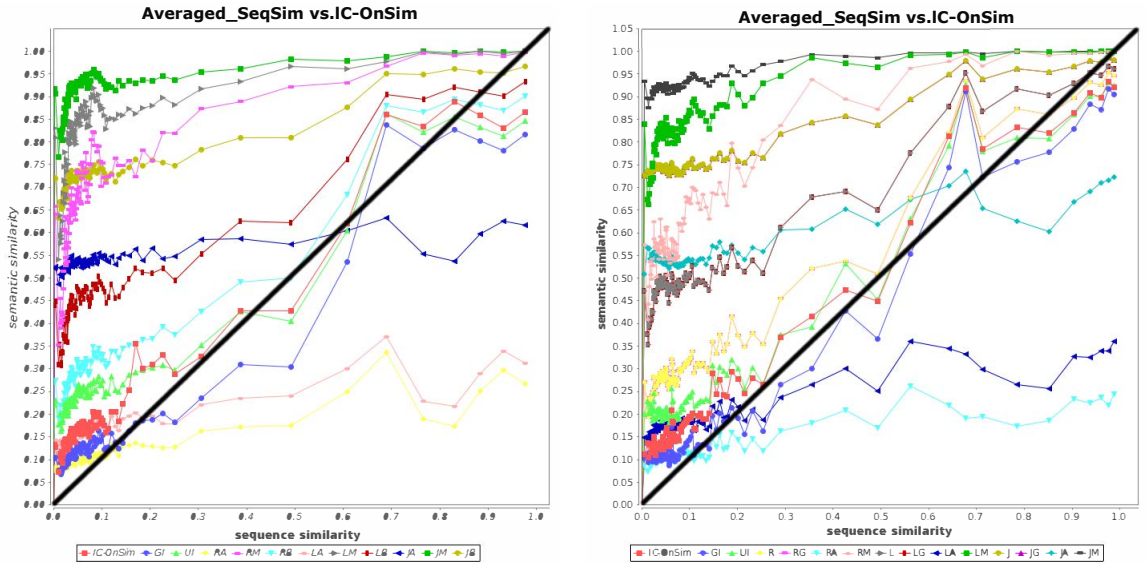
Table 4.5: Classification of the similarity measures in CESSM according to the considered resource characteristics.

measures estimate the similarity between two ontology terms based on the Information Content of their common ancestors. In order to aggregate term similarity values, the following strategies are considered. Similarity measures labeled with `A`, consider the average of the computed IC values. Those measures labeled with `M` follow the strategy proposed by Sevilla et al. [91] considering only the maximum value of IC. Measures labelled with `B` or `G` are aggregated term similarity values computing the best-match average of the ICs as proposed by Couto et al. [16]. Finally, the set-based measures simUI (UI) and simGIC (GI) [77] apply the Jaccard index to the annotation sets. In the case of GI the Jaccard coefficient is combined with the IC scores of the corresponding GO terms. Thus, UI only considers the GO term names in order to compute the similarity value, i.e., it does not consider any resource characteristic. OnSim and IC-OnSim are evaluated against these measures using the 1-1 maximum weighted matching [90] as aggregation function. Further, a baseline consisting of the combination of *AnnSim* [70], an implementation of the 1-1 maximum weighted matching with D_{tax} as similarity measure, is also evaluated on the benchmark. Table 4.5 summarizes the similarity measures included in CESSM and classifies them according to the considered resource characteristics. Figures 4.10(a)-(b) and 4.11(a)-(b) report on the comparison of *SeqSim* with OnSim, IC-OnSim and the above described measures.



(a) OnSim Pearson's Correlation with *SeqSim*:0.732-CESSM 2008. (b) OnSim Pearson's Correlation with *SeqSim*:0.772-CESSM 2014.

Figure 4.10: Results are produced by the CESSM tool for GO BP terms (versions 2008 and 2014) for OnSim.



(a) IC-OnSim Pearson's Correlation with *SeqSim*:0.776-CESSM 2008.

(b) IC-OnSim Pearson's Correlation with *SeqSim*:0.809-CESSM 2014.

Figure 4.11: Results produced by the CESSM tool for GO BP terms (versions 2008 and 2014) for IC-OnSim.

Empirical results: Plots in Figures 4.10(a) and 4.11(a) are generated on CESSM 2008 and Figures 4.10(b) and 4.11(b) on CESSM 2014. Results are produced by the CESSM tool for GO BP terms (versions 2008 and 2014). The similarity measures are: OnSim, IC-OnSim, simUI (UI), simGIC (GI), Resnik's Average (RA), Resnik's Maximum (RM), Resnik's Best-Match Average (RB/RG), Lin's Average (LA), Lin's Maximum (LM), Lin's Best-Match Average (LB), Jiang & Conrath's Average (JA), Jiang & Conrath's Maximum (JM), J. & C.'s Best-Match Average. (JB). The black diagonal lines in Figures 4.10(a)-(b) and 4.11(a)-(b) represent the value assigned by *SeqSim*. In almost all the cases, the studied similarity measures correctly detect when two proteins are similar in terms of *SeqSim* and assign high similarity values to these pairs of proteins, i.e., for proteins with high values of *SeqSim* the curves of the studied similarity measures are close to the black diagonal lines. However, the majority of these similarity measures do not exhibit the same level of accuracy distinguishing when two proteins are dissimilar, i.e., the corresponding curves are far from the black diagonal lines. Contrary, curves of IC-OnSim represented as red lines in the Figures 4.11(a) and (b), are closer to the black diagonal lines.

Table 4.6 summarizes the comparison of all the similarity measures with the gold standards: *ECC*, *Pfam*, and *SeqSim* on CESSM 2008 and 2014. It reports on Pearson's correlation coefficients, where the Top-5 values are highlighted in gray, and the highest correlation with respect to each of the baseline similarity measure is highlighted in bold. In the collection of 2008, IC-OnSim has the highest correlation with respect to *SeqSim* and *Pfam* (i.e., *0.776* and *0.537*, respectively) and it is the second most correlated function with respect to ECC. IC-OnSim is followed closely by simGIC (GI) [77] with *0.773* with respect to *SeqSim*. In comparison with GI, IC-OnSim additionally considers the taxonomic similarity, the neighborhood facts and the justifications that afford the entailment of these facts. Thus, a more precise estimate of the relatedness of two proteins is computed, i.e., both IC-OnSim and *SeqSim* assign low similarity values to a large number of pairs of proteins.

In the collection of 2014, IC-OnSim has the highest correlation with respect to *SeqSim* and *Pfam*, *0.809* and *0.491*, respectively, and it stays in the Top-5 most correlated measures with respect to *ECC*. OnSim does not distinguish so precisely dissimilar proteins, and the correlation with respect to *SeqSim* is lower in both datasets (*0.726* and *0.776* for 2008 and 2014, respectively). IC-OnSim enhances OnSim, and exhibits a performance similar to GI in dissimilar pairs of proteins, i.e., pairs of proteins with low *SeqSim* values; and a better performance in similar pairs of proteins. This improvement is the result of considering the information shared by the GO

terms, measured in terms of Information Content. This enhanced behavior also corroborates our hypothesis that IC-OnSim can positively impact on the effectiveness of annotation-based similarity measures.

Similarity measure	2008			2014		
	SeqSim	*ECC*	*Pfam*	*SeqSim*	*ECC*	*Pfam*
GI	0.773	0.398	0.454	0.799	0.458	0.421
UI	0.730	0.402	0.450	0.776	0.470	0.436
RA	0.406	0.302	0.323	0.411	0.308	0.264
RM	0.302	0.307	0.262	0.448	0.436	0.297
RB	0.739	**0.444**	0.458	0.794	**0.513**	0.424
LA	0.340	0.304	0.286	0.446	0.325	0.263
LM	0.254	0.313	0.206	0.350	0.460	0.252
LB	0.636	0.435	0.372	0.715	0.511	0.364
JA	0.216	0.193	0.173	0.517	0.268	0.261
JM	0.234	0.251	0.164	0.342	0.390	0.214
JB	0.586	0.370	0.331	0.715	0.451	0.355
AnnSim	0.650	0.388	0.459	0.682	0.434	0.407
OnSim	0.732	0.375	0.514	0.772	0.439	0.438
IC-OnSim	**0.776**	0.436	**0.537**	**0.809**	0.509	**0.491**

Table 4.6: Pearson's correlation coefficient with respect to the three gold standards of CESSM 2008 and CESSM 2014 results.

Performance Evaluation

In Section 4.2.3 the computational complexity of OnSim and IC-OnSim is introduced. In this section empirical experiments show the average performance of both measures in terms of time. The experiment consists on executing the two versions of the CESSM benchmark with OnSim and IC-OnSim. The experiment is divided in two phases: the initialization and the computation of the similarity values.

Table 4.7 shows the configuration of each performed experiment. The experiments are performed on CESSM 2008 and CESSM 2014 datasets. In CESSM 2008 13,430 annotated entities, proteins in this case, are compared. These entity comparisons require to compare 226,756 ontology terms. CESSM 2014 includes more than 22,000 entity comparisons and, therefore, the number of ontology term comparisons raises to 1,358,665. The increase of number of classes and axioms increments the time needed by the reasoner to classify the ontology and compute the neighborhoods. The Gene Ontology is one of the richest ontologies in terms of axioms. However, it is still within the OWL EL profile. The use of EL reasoners allows to classify OWL EL ontologies in polynomial time. In exchange, the expressiveness of the OWL EL profile is lower than the OWL 2 language. Further, in the OWL EL profile also the computation of justifications can be accelerated [112].

Odd configurations in Table 4.7 consider the full OWL2 semantics and use HermiT [29] as reasoner. Even configurations reduce the considered expressiveness to the OWL EL profile and use the ELK reasoner to take advantage of the properties of this profile. Additionally, even configurations use the method proposed by Zhou et al. [112] to compute inference justifications in OWL EL ontologies. All the experiments are performed on an Ubuntu 14.04 64bit machine with a CPU Intel(R) Core(TM) i5-4300U 1.9 GHz (4 physical cores) and 8GB RAM.

Observe the appreciable differences in terms of time among odd and even configurations in Table 4.8. The reduction of the expressiveness to the OWL EL profile allows to decrease the execution time of each phase. Table 4.9

ID	Dataset	Reasoner	Justification	Sim. Measure
1	2008	Hermit	Normal	OnSim
2	2008	ELK	Zhou [112]	OnSim
3	2014	Hermit	Normal	OnSim
4	2014	ELK	Zhou [112]	OnSim
5	2008	Hermit	Normal	IC-OnSim
6	2008	ELK	Zhou [112]	IC-OnSim
7	2014	Hermit	Normal	IC-OnSim
8	2014	ELK	Zhou [112]	IC-OnSim

Table 4.7: Configurations used for the empirical performance study of OnSim and IC-OnSim.

Config.	Initialization	Neighborhood	Similarity Matrix	Total time
1	1.09	7.778	0.13	8.998
2	0.46	0.035	0.11	0.605
3	2.11	25.067	2.349	29.526
4	1.03	0.429	0.633	2.092
5	1.02	7.42	0.175	8.615
6	0.52	0.035	0.15	0.705
7	1.75	20.848	2.14	24.738
8	1.26	0.435	1.068	2.763

Table 4.8: Times (minutes) corresponding to the three phases of the similarity measure.

shows the speedup coefficients according to the Amdahl's law [3]. The coefficient indicates the number of times the even configurations can be executed while the odd configurations complete their first execution. Each coefficient is the average among the experiments in CESSM 2008 and 2014. The computation of the neighborhood is the phase that exhibits the best improvement due to the method applied to obtain the justifications with a speedup of 140.33 and 129.96 for OnSim and IC-OnSim respectively.

Measure	Phase	Speedup
OnSim	Initialization	2.21
OnSim	Neighborhood	140.33
OnSim	Similarity matrix	2.45
IC-OnSim	Initialization	1.68
IC-OnSim	Neighborhood	129.96
IC-OnSim	Similarity matrix	1.59

Table 4.9: Average speedup coefficients of the three phases of OnSim and IC-OnSim

4.3 Semantic Similarity Measures on Graph Structured Data: GADES

Semi-structured data is a form of structured data that does not follow the classical structure of relational data models like tables, but has some organizational properties to facilitate its processing. Graphs are becoming a popular way of representing this kind of data. For example, initiatives like DBpedia [5] or YAGO [59] provide semi-structured representations of the different versions of the Wikipedia in form of knowledge graphs. The use of new structures to represent data requires the definition of similarity measures able to deal with the new representations.

In this section GADES, a similarity measures for graph structured data is defined. GADES meets the metric properties and considers four resource characteristics. Similarity values of each resource characteristic are combined to obtain the final similarity value according to aggregation functions defined by users or domain experts.

OnSim and IC-OnSim exhibit good performance in CESSM benchmarks in terms of correlation with gold standards. However, they have some disadvantages. First, their computational complexity is high, so they are not suitable for near real-time use cases. Second, they do not meet the metrics property due to the 1-1 MWBM performed between term neighborhoods, which impedes its use in clustering algorithms. Finally, they do not consider literals, i.e., numeric or string attributes of the nodes in the knowledge graph. In this section GADES, a neiGhborhood-bAseD graph Entity Similarity, is introduced . GADES considers the knowledge encoded in all *resource characteristics*, i.e., hierarchies, neighborhoods, shared information, and datatype properties or *attributes*. GADES receives as input a knowledge graph and two entities to be compared. As a result, GADES outcomes a similarity value that aggregates similarity values computed in terms of *resource characteristics*. GADES is evaluated comparing entities of four different knowledge graphs. Two knowledge graphs describe texts (news articles and video descriptions) with DBpedia entities. The other two knowledge graphs describe proteins with GO entities and correspond to the 2008 and 2014 versions of CESSM. GADES is compared with state-of-the-art similarity measures and outperforms them in terms of correlation with respect to the provided gold standards.

The work included in this section is published in the proceedings of the *SEMANTiCS* Conference in 2015 [105].

4.3.1 Motivating Example

Figure 4.12 contains a portion of a knowledge graph describing countries, international agreements and organizations together with their relations and attributes. For the sake of clarity edges have different colours to better distinguish the origin and the target. Each entity in the knowledge graph is related to others through different relations or RDF *properties*, e.g., `hasMember`, `signedBy`, etc. These properties are also described in terms of the RDF property `rdf:type` as depicted in Figure 4.12. The relatedness between entities in this knowledge graph is determined based on different *resource characteristics*, i.e., hierarchy, neighborhood, specificity, and attributes. Consider entities *CETA*, *NATO* and *UN Security Council*. Hierarchies in the knowledge graph represented in Figure 4.12 are induced by the *rdf:type* RDF property, which describes an entity as instance of an RDF class.

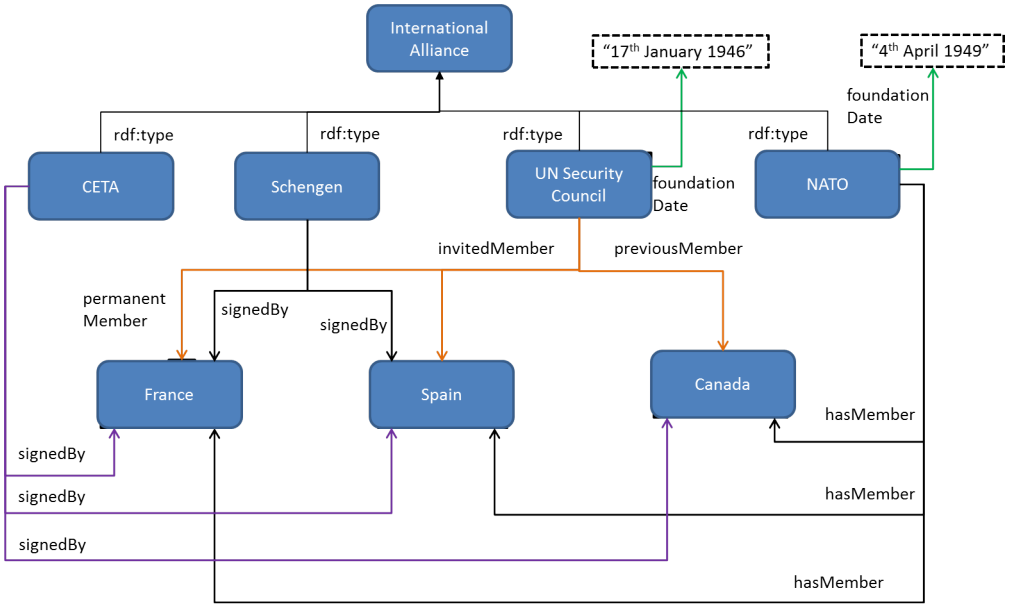


Figure 4.12: Motivating Example. Portions of knowledge graph describing countries and international alliances.

Particularly, these entities are described as instances of the *International Alliances* class, which is supposed to be at a deep level in the hierarchy. Thus, according to the hierarchy, these entities are highly similar.

Further, these entities share exactly the same set of neighbors that is formed by the countries *Spain*, *France*, and *Canada*. However, the relations among the international alliances and the countries are different for each case. *CETA* is related to the three countries through the property `signedBy`. *NATO* and *UN Security Council* are related to the three countries through the property `hasMember` or one of its sub-properties according to the property hierarchy in Figure 4.13. Considering only the entities contained in these neighborhoods, the compared international alliances are identical since they share exactly the same neighbors. However, whenever RDF properties, *NATO* and *UN Security Council* are more similar since both organizations are related to *France*, *Spain* and *Canada* with some sub-property of *hasMember*, while the relation among *CETA* and these countries is `signedBy`.

Additionally, international alliances are also related with attributes through datatype properties. For the sake of clarity only a portion of these attributes is included in Figure 4.12. Considering these attributes, *UN Security Council* was founded on January, 17th 1946, while *NATO* was founded on April, 4th 1949.

Finally, the shared information is different for each entity in the graph. The more frequently two entities co-occur in a neighborhood, the more information they share. In Figure 4.12 the entities *Spain* and *France* are contained in the neighborhoods of four entities, while *Spain* and *Canada* only co-occur in three neighborhoods. Thus, the information shared by *Spain* and *France* is higher than the information shared among *Spain* and *Canada*.

These observations suggest that the similarity between two knowledge graph entities does not depend on only one *resource characteristic*, and that combinations of them may have to be considered to precisely determine relatedness between entities in a knowledge graph.

4.3.2 GADES Architecture

GADES is a semantic similarity measure for comparing entities in knowledge graphs. GADES considers the knowledge encoded in the four identified *resource characteristics*, i.e., hierarchies, neighborhoods, shared information, and attributes to accurately determine relatedness between entities in a knowledge graph. GADES computes values of similarity for each *resource characteristic* independently and combines the comparison results to produce an aggregated similarity value between the compared entities. Figure 4.14 depicts the architecture of

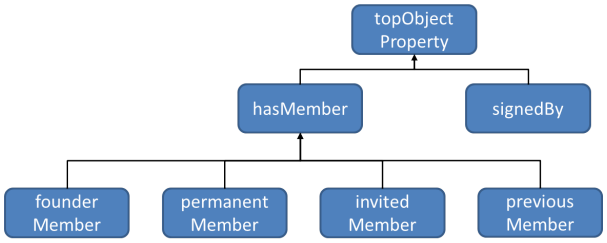


Figure 4.13: Property Hierarchy. Portion of a knowledge graph describing a relation or RDF property hierarchy.

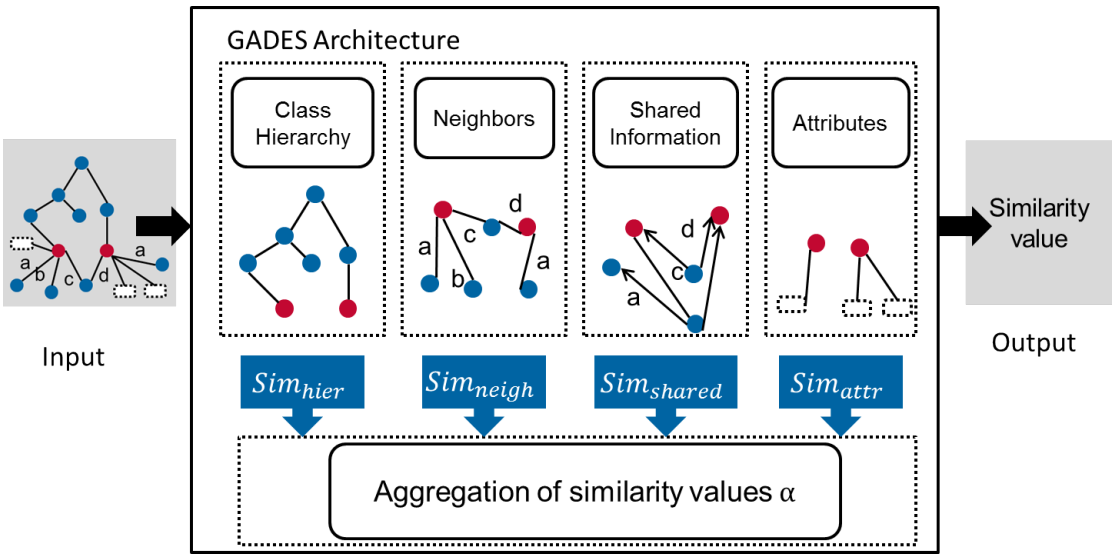


Figure 4.14: GADES Architecture.

GADES. GADES receives as input a knowledge graph G and two entities e_1, e_2, represented as red nodes, to be compared. GADES computes similarity values based on the different *resource characteristics*, i.e., the hierarchy in the knowledge graph, the neighborhoods, the shared information and the attributes of the given entities. Produced similarity values are aggregated with an aggregation function α, which combines individual similarity measures $Sim_{hier}, Sim_{neigh}, Sim_{shared}, Sim_{attr}$. The aggregated value is returned as output.

Definition 11. ***Individual similarity measure.*** *Given a knowledge graph $G = (V, E, L)$, two entities e_1 and e_2 in V, and a resource characteristic RC of e_1 and e_2 in G, an individual similarity measure $Sim_{RC}(e_1, e_2)$ corresponds to a similarity function defined in terms of RC for e_1 and e_2.*

GADES combines individual similarity measures to produce a similarity value using an aggregated similarity measure α. Particularly, GADES combines four individual similarity measures: the hierarchical similarity $\text{Sim}_{\text{hier}}(e_1, e_2)$ or the neighborhood similarity $\text{Sim}_{\text{neigh}}(e_1, e_2)$, the shared information similarity $\text{Sim}_{\text{shared}}(e_1, e_2)$ and the attribute similarity $\text{Sim}_{\text{attr}}(e_1, e_2)$.

Definition 12. ***Aggregated similarity measure.*** *Given a knowledge graph $G = (V, E, L)$ and two entities e_1 and e_2 in V. An aggregated similarity measure α for e_1 and e_2 is defined as follows:*

$$GADES(e_1, e_2) = \alpha(e_1, e_2 | \top, \beta, \gamma) = \top(\beta(e_1, e_2), \gamma(e_1, e_2)),$$

where:

- $\top$ *is a triangular norm (T-Norm) [46],*

- $\beta(e_1, e_1)$ *and* $\gamma(e_1, e_2)$ *are aggregated or individual similarity measures.*

GADES corresponds to an aggregated similarity measure α, which depends on the application domain. α combines individual similarity measures relying on resource characteristics, e.g., hierarchies, neighborhoods, specificity, and attributes. Next, the four individual similarity measures considered by GADES are described.

Hierarchical similarity Given a knowledge graph G, the hierarchy is inferred by the set of hierarchical edges HE. Hierarchical edges $HE = \{(v_i, r, v_j) | (v_i, r, v_j) \in E \wedge \text{Hierarchical}(r)\}$ are a subset of knowledge graph edges whose property names refer to a hierarchical relation, e.g., *rdf:type*, *rdfs:subClassOf*, or *skos:broader*. Generally, every relation that introduces an entity as a generalization (ancestor) or an specification (successor) of another entity is a hierarchical relation. GADES uses hierarchical similarity measures as D_{tax} [10] and D_{ps} [75] to measure the hierarchical similarity between two entities. Both measures are based on the LCA intuition: similar entities have a deep and close lowest common ancestor.

$$\mathbf{Sim_{hier}}(e_1, e_2) = \begin{cases} 1 - D_{\text{tax}}(e_1, e_2) \\ 1 - D_{\text{ps}}(e_1, e_2) \end{cases}$$

Neighborhood similarity The neighborhood of an entity $e \in V$ is defined as the set of relation-entity pairs $N(e)$ whose entities are at one-hop distance of e, i.e., $N(e) = \{(r, e_i) | (e, r, e_i) \in E\}$. This definition of neighborhood allows for considering together the neighbor entity and the relation type of the edge. GADES uses the knowledge encoded in the relation and class hierarchies of the knowledge graph to compare two pairs $p_1 = (r_1, e_1)$ and $p_2 = (r_2, e_2)$. The similarity between two pairs p_1 and p_2 is computed as $\text{Sim}_{\text{pair}}(p_1, p_2) = \text{Sim}_{\text{hier}}(e_1, e_2) \cdot \text{Sim}_{\text{hier}}(r_1, r_2)$. Note that Sim_{hier} can be used with any resource of the knowledge graph, regardless of wether it is an instance, a class or a relation. In order to aggregate the similarity between two neighborhoods, GADES combines pair comparisons using the metric defined by Fujita [25]:

$$\mathbf{Sim_{neigh}}(e_1, e_2) = 1 - \frac{1}{|N(e_1)||N(e_2)|} \left(\sum_{p_i \in N(e_1)} \sum_{p_j \in N(e_2)} (1 - \text{Sim}_{\text{pair}}(p_i, p_j)) - \sum_{p_i \in N(e_1) \cap N(e_2)} \sum_{p_j \in N(e_1) \cap N(e_2)} (1 - \text{Sim}_{\text{pair}}(p_i, p_j)) \right)$$

In Figure 4.12, the neighborhoods of *UN Security Council* and *NATO* are {(`permanentMember`, *France*), (`invitedMember`, *Spain*), (`previousMember`, *Canada*)} and {(`hasMember`, *France*), (`hasMember`, *Canada*), (`hasMember`, *Spain*)}, respectively. Let $\text{Sim}_{\text{hier}}(e_1, e_2) = 1 - D_{\text{tax}}(e_1, e_2)$. The most similar pair to (`permanentMember`, *France*) is (`hasMember`, *France*) with a similarity value of 0.67. This similarity value is result of the product between Sim_{hier}(*France*, *France*), whose result is 1.0, and Sim_{hier}(`permanentMember`, `founderMember`), whose result is 0.5. In a like manner, the most similar pair to (*invitedMember*, *Spain*) is (*hasMember*, *Spain*) with a similarity value of 0.67. Finally, the most similar pair to (`previousMember`, *Canada*) is the pair (`hasMember`, *Canada*) with a similarity value of 0.67. Thus, the similarity between neighborhoods of *UN Security Council* and *NATO* is computed as $\text{Sim}_{\text{neigh}} = \frac{(0.67+0.67+0.67)+(0.67+0.67+0.67)}{3+3} == 0.67$.

Shared Information Beyond the hierarchical similarity, the amount of information shared by two entities in a knowledge graph can be measured examining the human use of such entities. Two entities are considered to share a lot of information if they are similarly used in a corpus. Considering the knowledge graph as a corpus, the information shared by two entities x and y is directly proportional to the amount of entities that have x and y together in their neighborhood, i.e., the co-occurrences of x and y in the neighborhoods of the entities.

Let $G = (V, E, L)$ be a knowledge graph and $e \in V$ an entity in the knowledge graph. The set of entities that have e in their neighborhood is defined as $Incident(e) = \{e_i \mid (e_i, r, e) \in E\}$. Then, GADES computes the information shared by two entities using the following formula:

$$\mathbf{Sim_{shared}}(e_1, e_2) = \frac{|\text{Incident}(e_1) \cap \text{Incident}(e_2)|}{|\text{Incident}(e_1) \cup \text{Incident}(e_2)|},$$

where the denominator of this formula helps decreasing the similarity with respect to abstract or not informative entities. For example, in a knowledge graph like DBpedia the entity representing *Germany* is included in several neighborhoods. This means that *Germany* is not a specific entity, so a co-occurrence with another entity should not be considered equally important than the co-occurrence of two specific entities.

In Figure 4.12 entities *France*, *Spain* and *Canada* have incident edges. *France* and *Spain* have four incident edges, while *Canada* has only three. *France* and *Spain* co-occur in three of the four neighborhoods. Particularly, both countries are related with *Schengen* and *CETA* through the property `signedBy`. Further, *France* and *Spain* are described as members of the NATO. Thus, $\text{Sim}_{\text{shared}}$ returns a value of $\frac{3}{4} = 0.75$. *Canada* co-occurs with both *France* and *Spain* in two neighborhoods. Thus, $\text{Sim}_{\text{shared}}$ returns a value of $\frac{2}{4} = 0.5$.

Attribute Similarity Entities in knowledge graphs are not only related with other entities, but also with attributes through datatype properties, e.g., temperature, protein sequence, etc. GADES considers only shared attributes, i.e., attributes connected to entities through the same datatype property. Given that attributes can be compared with domain similarity measures, e.g., SeqSim [78] for gene sequences or Manhattan or Euclidean distance for vectors, GADES does not describe a specific measure to compare all kind of attributes. Depending on the domain, experts will choose a similarity measure for each type of attribute.

In Figure 4.12 entities *UN Security Council* and *NATO* have the attribute `foundationDate`. Thus, Sim_{attr} between these entities would consider this attribute to determine the similarity. In case there were more attributes, only common attributes of the compared entities will be taken into account during the computation of Sim_{attr}.

4.3.3 Theoretical Properties

One of the research question of this thesis is about the scalability of similarity measures in large datasets. This section includes a study about the computational complexity of GADES. This computational complexity study is accompanied by an empirical study on performance in Section 4.3.4. Further, for the sake of clarity, a proof of the metric properties for GADES is shown in Appendix A.

Time Complexity

GADES takes into account only the explicit knowledge in the knowledge graph, i.e., only the explicitly described edges are considered. Thus, the initialization phase is not required anymore. Further, GADES implements the aggregation metric proposed by Fujita [25] instead of the 1-1 maximum matching to compute the neighborhood similarity. Finally, GADES also considers the attributes when computing the similarity. The added complexity will depend on the complexity of the attribute similarity measure.

Let $G = (V, E, L)$ be the knowledge graph containing the entities to be compared. Let $V' \subseteq V$ be the set of knowledge graph entities to be pairwise compared. Before performing the pairwise comparison, the neighborhood of each term is computed. Given that only explicit knowledge is considered, the computation of the neighborhood in the worst case is constant time $O(1)$. Therefore, the complexity of computing $|V'|$ neighborhoods is $O(|V'|)$.

GADES considers four *resource characteristics*: the hierarchy, the neighborhood, the shared information and the attribute similarity. Thus, the computational complexity of GADES is described as:

Function	Complexity
Initialization	$O(\|V'\|)$
$\text{Sim}_{\text{neigh}}$	$O(\|V\|^{4.575} \cdot \|L\|^{2.575})$
$\text{Sim}_{\text{shared}}$	$O(\|V\|)$
Sim_{attr}	$O(\|V\|^2 \text{Sim}_{\text{string}})$
GADES	$O(\|V\|^{4.575} \cdot \|L\|^{2.575} + \|V\|^2 \text{Sim}_{\text{string}})$

Table 4.10: Computational complexity of each function computed by GADES

$$O(\text{GADES}) = O(\text{Sim}_{\text{hier}}) + O(\text{Sim}_{\text{neigh}}) + O(\text{Sim}_{\text{shared}}) + O(\text{Sim}_{\text{attr}})$$

Like OnSim and IC-OnSim, Sim_{hier} is implemented in GADES with D_{tax} or D_{ps}. As showed in Section 4.2.3, the computational complexity of these measures is equivalent to the complexity of computing the LCA of the compared entities. Thus, the complexity of Sim_{hier} is $O(|V|^{2.575})$ [33].

Algorithm 4 describes the function $\text{Sim}_{\text{neigh}}$ for GADES, which corresponds to the aggregation metric defined by Fujita [25]. In the worst case the compared entities have disjoint sets of neighborhoods with size $|V|/2$, i.e., $N_1 \cap N_2 \equiv \emptyset$ and $|N_1| = |N_2| = |V|/2$. Thus, the computational complexity of $\text{Sim}_{\text{neigh}}$ is described as follows:

$$O(\text{Sim}_{\text{neigh}}) = O(\frac{|V|^2}{2} \text{Sim}_{\text{pair}})$$

Sim_{pair} requires the computation of two Sim_{hier} values: the hierarchical similarity between the relations and the hierarchical similarity between the target entities. Then, the computational complexity of Sim_{pair} is defined as:

$$O(\text{Sim}_{\text{pair}}) = O(|V|^{2.575} \cdot |L|^{2.575})$$

$\text{Sim}_{\text{shared}}$ computes the intersection and the union of the incident edges of the compared entities. In the worst case every entity in the graph $v \in V$ has an edge to the compared entities. Because of both operations can be implemented with linear complexity, the complexity of both operations is $O(|V|)$.

Finally, Sim_{attr} aggregates attribute similarity values using the metric defined by Fujita [25]. Attributes in a knowledge graph can be numbers or strings. Numbers can be compared in constant time by means og mathematical operations like difference or division. There exist several string metrics in the state of the art to compare strings. For example, the Jaro-Winkler [40] distances has a complexity of $O(l^2)$, where l is the maximum length of the compared strings. Then, the computational complexity of Sim_{attr} in the worst case is described as follows:

$$O(\text{Sim}_{\text{attr}}) = O(\frac{|V|^2}{2} \text{Sim}_{\text{string}}),$$

Table 4.10 summarizes the computational complexity of each of the functions computed by GADES. The computational complexity of GADES is $O(|V|^{4.575} \cdot |L|^{2.575} + |V|^2 \text{Sim}_{\text{string}})$. Thus, GADES has a lower computational complexity than OnSim and IC-OnSim, both $O(|V|^3|E|^2 d + |V|^2|E|^3)$, if the knowledge graph does not contain isolated nodes, i.e., $|E| \geq \frac{|V|}{2}$.

Proof of Metric

GADES meets the properties of a metric; therefore,GADES can be consistently used with clustering algorithms. For the sake of clarity, the proof of the metric properties is detailed in Appendix A.

Algorithm 4 $\text{Sim}_{\text{neigh}}$

```
1: procedure SIM_NEIGH(a_1, a_2)
2:     N_1 ← Neighbors(a_1)
3:     N_2 ← Neighbors(a_2)
4:     sim_1 = 0
5:     for p_x ∈ N_1 do
6:         for p_i ∈ N_2 \ N_1 do
7:             sim_1 += 1 − Sim_pair(p_x, p_i)
8:         end for
9:     end for
10:    sim_1 = sim_1/(|N_1 ∪ N_2||N_1|)
11:    sim_2 = 0
12:    for p_y ∈ N_2 do
13:        for p_i ∈ N_1 \ N_2 do
14:            sim_2 += 1 − Sim_pair(p_y, p_i)
15:        end for
16:    end for
17:    sim_2 = sim_2/(|N_1 ∪ N_2||N_2|)
18:    return sim = 1 − sim_1 − sim_2
19: end procedure
```

4.3.4 Empirical Properties

In this section the empirical properties of GADES are shown. First, GADES is evaluated on CESSM 2008 and 2014 and compared to OnSim and IC-OnSim. Next, GADES is evaluated on two datasets of texts annotated with DBpedia entities and compared with state-of-the-art similarity measures. Finally, an empirical study on performance confirms that GADES is faster than OnSim and IC-OnSim and is suitable for near real-time applications.

Evaluation on Ontology-based Annotated Datasets

The behavior of GADES is studied using CESSM 2008 and 2014 with the 1-1 maximum weighted matching as aggregation strategy. The same configuration is used for OnSim and IC-OnSim in section 4.2.4 is followed in this study. Table 4.11 extends Table 4.6 including the results for GADES. Table 4.11 contains the Pearson's correlation coefficient between three gold standards and eleven similarity measures of CESSM. The Top-5 correlations are highlighted in gray, and the highest correlation with respect to each gold standard is highlighted in *bold*. These results suggest that inferred relations and their justifications are not necessary to compute accurate similarity values. GADES is competitive with respect to the other measures being one of the top-5 most correlated measures with the gold standards. In both version of CESSM, the different resource characteristic similarities are combined with the following functions:

$$\alpha_1(e_1, e_2|\top_1, \text{Sim}_{\text{hier}}, \text{Sim}_{\text{neigh}}) = \top_1(\text{Sim}_{\text{hier}}, \text{Sim}_{\text{neigh}}),$$

where $\top_1(a,b) = a \cdot b$ is the product T-Norm and $\text{Sim}_{\text{hier}} = 1 - d_{\text{tax}}$.

$$\alpha_2(e_1, e_2|\top_1, \alpha_1(e_1, e_2), \text{Sim}_{\text{shared}}) = \top_1(\alpha_1(e_1, e_2), \text{Sim}_{\text{shared}}),$$

where α_2 corresponds to GADES.

Similarity	2008			2014		
measure	*SeqSim*	*ECC*	*Pfam*	*SeqSim*	*ECC*	*Pfam*
GI	0.773	0.398	0.454	0.799	0.458	0.421
UI	0.730	0.402	0.450	0.776	0.470	0.436
RA	0.406	0.302	0.323	0.411	0.308	0.264
RM	0.302	0.307	0.262	0.448	0.436	0.297
RB	0.739	**0.444**	0.458	0.794	0.513	0.424
LA	0.340	0.304	0.286	0.446	0.325	0.263
LM	0.254	0.313	0.206	0.350	0.460	0.252
LB	0.636	0.435	0.372	0.715	0.511	0.364
JA	0.216	0.193	0.173	0.517	0.268	0.261
JM	0.234	0.251	0.164	0.342	0.390	0.214
JB	0.586	0.370	0.331	0.715	0.451	0.355
D_{tax}	0.650	0.388	0.459	0.682	0.434	0.407
OnSim	0.732	0.375	0.514	0.772	0.439	0.438
IC-OnSim	**0.776**	0.436	0.537	0.809	0.509	**0.491**
GADES	0.773	0.437	**0.546**	**0.836**	**0.515**	0.49

Table 4.11: CESSM 2008 and CESM 2014 results.

Evaluation on Graph-based Data

GADES is also evaluated on graph-based data. Particularly, GADES is evaluated on two datasets, Lee50 and STS, that describe real world entities, in this case news articles and sentences, with DBpedia entities. Both datasets are considered standard benchmarks for the text comparison tasks according to several works [72, 1, 26]. The DBpedia knowledge graph is larger than the GO knowledge graph. Unlike GO, DBpedia contains information automatically extracted from the web, particularly DBpedia. Hence, DBpedia is prone to contain more noisy data than GO, which is only manually curated.

Lee50: News Articles Comparison The Lee50 [54] dataset is considered a standard benchmark for the comparison of multiple sentence documents and have been used in several works to proof their efficiency [51, 35, 88, 26, 72]. The benchmark contains 50 news articles pairwise compared. Each article has a length among 51 and 126 words. The similarity of each pair of news articles is estimated by several humans, rating each comparison with values among 1 (poorly similar) and 5 (highly similar). The evaluation of the similarity of each document pair by multiple humans allows to moderate the subjective perspectives of such humans. In order to have one similarity value for each document pair, human ratings are aggregated using the average. The quality of each similarity measure is computed as the Pearson's correlation coefficient with respect to the human-based similarity values. The Lee50 dataset was used by Paul et al. [72] to evaluate GBSS, a semantic similarity measure able to consider the neighborhood and the hierarchy at the same time to compute similarity values. Paul et al. enriched the Lee50 dataset by annotating each news article with DBpedia entities. Each article receives on average 10 annotations.

GADES considers three of the four *resource characteristics* in the Lee50: the hierarchy, the neighborhood and the shared information. Attributes or literals are not considered during the evaluation of this dataset. DBpedia literals provide very specific information about the entity, e.g. birth date of a person, height or even Wikipedia IDs. Therefore, it is assumed that this kind of information is not relevant when comparing two news articles according

to the identified entities on them. The aggregation functions α_1 and α_2 serve to combine the three similarity values and are defined as follows:

$$\alpha_1(e_1,e_2|\top_1,\text{Sim}_{\text{hier}},\text{Sim}_{\text{shared}}) = \top_1(\text{Sim}_{\text{hier}}(e_1,e_2),\text{Sim}_{\text{shared}}(e_1,e_2)),$$

where $\top_1$ corresponds to the product T-Norm and $\text{Sim}_{\text{hier}} = 1 - D_{\text{tax}}$

$$\alpha_2(e_1,e_2|\top_2,\alpha_1,\text{Sim}_{\text{neigh}}) = \top_2(\alpha_1(e_1,e_2),\text{Sim}_{\text{neigh}}(e_1,e_2)),$$

where $\top_2(a,b) = \frac{a+b}{2}$

Table 4.12 summarizes the Pearson's correlation coefficient of GADES and state-of-the-art similarity measures. Similarity measures that do not consider information encoded in knowledge graphs, e.g. LSA, SSA, GED or ESA, obtain lower correlation values than GADES or GBSS, which combine several resource characteristics to determine similarity values. Further, GADES is able to outperform GBSS. GBSS only considers two resource characteristics: the hierarchy and the neighborhood. Thus, the empirical results show that the information encoded in the knowledge graph is relevant for the computation of similarity values and that combining multiple resource characteristics allows to improve the accuracy of the similarity measures.

Similarity measure	**Pearson's coefficient**
LSA [51]	0.696
SSA [35]	0.684
GED [88]	0.63
ESA [26]	0.656
D_{ps} [75]	0.692
D_{tax} [10]	0.652
$\text{GBSS}_{r=1}$ [72]	0.7
$\text{GBSS}_{r=2}$ [72]	0.714
$\text{GBSS}_{r=3}$ [72]	0.704
GADES	**0.727**

Table 4.12: Lee50 results. Pearson's coefficient for GADES and state-of-the-art measures in the Lee et al. knowledge graph [54].

Similarity measure	**Pearson's coefficient**
Polyglot [2]	0.696
Tiantian. [113]	0.684
IRIT [12]	0.63
D_{ps} [75]	0.703
D_{tax} [10]	**0.71**
$\text{GBSS}_{r=2}$ [72]	0.666
$\text{GBSS}_{r=3}$ [72]	0.673
GADES	**0.71**

Table 4.13: STS results. Pearson's coefficient for GADES and state-of-the-art measures in the 2012-MSRvid-Test

STS: Sentence Comparison The second dataset called STS corresponds to the SemEval-2012[11] shared task. This dataset was published in 2012 aiming to provide researchers of a benchmark to measure the semantic similarity among sentences. Similarly to Lee50, the STS dataset has also been used in different works to evaluate the quality of several similarity measures [2, 113, 12, 72]. The benchmark consists of a set of 750 sentence comparisons and a similarity gold standard based on a human evaluation. Sentences describe events in videos and have a length among 8 and 10 words. The quality of a similarity measure is computed as the Pearson's coefficient with respect to the human-based gold standard. The STS dataset was also used to evaluate GBSS [72]. Paul et al. [72] enriched 689 of these sentences with DBpedia instances in the same way they did with the Lee50 dataset. Each sentence is annotated with among 1 and 3 DBpedia instances.

[11] https://www.cs.york.ac.uk/semeval-2012/task6/index.html

Dataset	Neighborhood	Similarity Matrix	Total time
2008	0.021	0.119	0.14
2014	0.169	0.616	0.785

Table 4.14: Times (minutes) corresponding to the two phases of the similarity measure.

The low number of knowledge graph entities describing each resource (among 1 and 3) is not an enough large corpus to determine the shared information among the entities. Thus, GADES considers only the hierarchy and the neighborhoods during the comparison of sentences in the STS dataset.

Table 4.13 shows that GADES is able to outperform state-of-the-art similarity measures in terms of correlation obtaining a Pearson's correlation coefficient of 0.71. GADES combines the hierarchical and neighborhood knowledge with the following function:

$$\alpha(e_1,e_2|\top,\text{Sim}_{\text{hier}},\text{Sim}_{\text{neigh}}) = \top_1(\text{Sim}_{\text{hier}}(e_1,e_2),\text{Sim}_{\text{neigh}}(e_1,e_2))\,,$$

$$\text{where } \top(a,b) = \tfrac{a+b}{2} \text{ and } \text{Sim}_{\text{hier}} = 1 - D_{\text{tax}}$$

Observe that D_{tax} ties with GADES. This observation is justified with the number of neighbors per entity. On average, each entity in this knowledge graph has 1.21 neighbors, while the average in the previous knowledge graph (Lee50) is almost the double (3.43 neighbors/entity). The lack of neighbors impedes GADES to show significant benefits when considering the neighborhood as relevant *resource characteristic*.

Performance Evaluation

In this section, empirical experiments show the average performance of GADES in terms of time. The experiment consists of executing the two versions of the CESSM benchmark with GADES.

GADES does not require the execution of any reasoner, so there is no initialization phase. Table 4.14 contains the execution times obtained by GADES in CESSM 2008 and 2014. Observe that GADES is faster than OnSim and IC-OnSim. OnSim and IC-OnSim required 0.605 and 0.705 minutes to execute CESSM 2008, respectively, GADES only needs 0.14. This means a speedup of 4.32 and 5.04 respectively. Similarly, in CESSM 2014 OnSim and IC-OnSim needs 2.092 and 2.763 minutes, while GADES just needs 0.785. The respective speedup coefficients are 2.66 and 3.52.

4.4 Automatic Aggregation of Individual Similarity Measures: GARUM

GADES is able to consider the four identified resource characteristics through their corresponding individual similarity measures Sim_{hier}, $\text{Sim}_{\text{neigh}}$, $\text{Sim}_{\text{shared}}$ and Sim_{attr}. However, the aggregation function of these individual similarity measures has to be defined by the user or a domain expert. GARUM[12] is a semantic similarity measure able to estimate the relevance of each resource characteristic depending on the domain. GARUM makes use of supervised machine learning approaches, where the combination of individual similarity measures is defined as a regression problem. Thus, the supervised machine learning algorithm learns, with the help of training data, how to combine the individual similarity measures to obtain an accurate similarity value.

[12] In the ancient Roma, it was necessary to preserve food for a long time. Romans created and refined the receipt of GARUM, a specific dressing to preserve and cook fish. The recipe of GARUM was found at the end of 2014 and explain how much of each ingredient is necessary to maintain adequately the fish. In the case of the presented similarity measures, a *recipe* is also needed to combine adequately each individual similarity measure.

4.4.1 Motivating Example

In Section 4.3.4 GADES is used to compare resources in four different knowledge graphs: CESSM 2008, CESSM 2014, Lee50, and STS. In the first two knowledge graphs, the individual similarities Sim_{hier}, $\text{Sim}_{\text{neigh}}$ and $\text{Sim}_{\text{shared}}$ are combined with a product triangular norm:

$$\text{GADES}(e_1, e_2) = \top_1(\alpha_1(e_1, e_2), \text{Sim}_{\text{shared}}),$$

where $\top_1(a,b) = a \cdot b$ is the product T-Norm and α_1 is defined as follows:

$$\alpha_1(e_1, e_2 | \top_1, \text{Sim}_{\text{hier}}, \text{Sim}_{\text{neigh}}) = \top_1(\text{Sim}_{\text{hier}}, \text{Sim}_{\text{neigh}}),$$

In the experiment over Lee50, three individual measures are combined with a product triangular norm and the average:

$$\text{GADES}(e_1, e_2) = \top_2(\alpha_1(e_1, e_2), \text{Sim}_{\text{neigh}}(e_1, e_2)),$$

where $\top_2$ is the average or mean function and α_1 is defined as follows:

$$\alpha_1(e_1, e_2 | \top_1, \text{Sim}_{\text{hier}}, \text{Sim}_{\text{shared}}) = \top_1(\text{Sim}_{\text{hier}}(e_1, e_2), \text{Sim}_{\text{shared}}(e_1, e_2)),$$

where $\top_1(a,b) = a \cdot b$

The STS dataset consists of sentences annotated with DBpedia entities. Hence, the setting is similar to the Lee50 dataset, which instead of sentences contains news articles annotated with DBpedia entities. Therefore, the combination of the individual similarity measures should be the same. However, the STS dataset is much smaller than Lee50. While Lee50 contains 1275 comparisons among news articles annotated on average with 10 DBpedia entities, STS only contain 689 comparisons among sentences annotated with few entities (among 1 and 3). Thus, the size of the dataset does not allow to compute $\text{Sim}_{\text{shared}}$ properly, which is based on the amount of common incident edges, i.e., the co-occurrence of knowledge graph resources in annotation datasets. For this reason the combination of individual similarity measures in STS is the same combination as in Lee50, but removing the $\text{Sim}_{\text{shared}}$ individual similarity. This may mean that the weight of this measure in this graph is 0. Therefore, for each dataset a different combination of the same components is necessary. Finding the best combination for each knowledge graph can be a tedious and time costly task. Hence, a method to automatically compute these combinations may help stakeholders to use GADES as similarity measure in their datasets.

4.4.2 GARUM: A Machine Learning Based Approach

Supervised machine learning algorithms infer functions from labeled training data. Training data consists of pairs (input, output); the input is usually a feature vector and the output can be a number or a category. When the output is a number, the problem solved by the supervised machine learning algorithm is called a regression problem. Otherwise, it is called a classification problem. In regression problems, supervised machine learning algorithms learn how to combine the input features to produce a value close to the defined output. Finding the best combination of individual similarity measures can be considered as an instance of a regression problem. GARUM is a semantic similarity measures that makes use of machine learning algorithms to determine the relevance of individual similarity measures depending on the domain to deliver accurate similarity values. Figure 4.15 shows the architecture of GARUM. GARUM receives a knowledge graph G and two entities, represented as red nodes, to be compared. Based on the information encoded in the knowledge graph, GARUM computes similarity values based on the class hierarchy in the knowledge graph, the neighborhoods, the specificity and the attributes of the given entities. Produced similarity values are aggregated by means of a machine learning algorithm that estimates the relevance

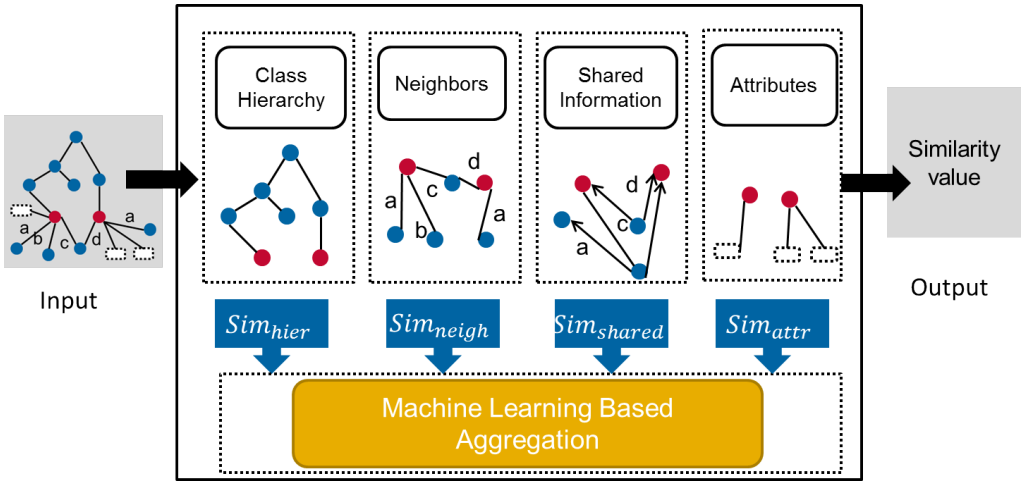


Figure 4.15: GARUM Architecture.

of individual similarity measures $Sim_{hier}, Sim_{neigh}, Sim_{shared}, Sim_{attr}$. The aggregated value is returned as output. Observe that the only difference with respect to GADES relies on the aggregation of individual similarity values. Promising machine learning algorithms for GARUM are neural networks and Support Vector Regression [18].

GARUM as a Regression Problem

A regression problem consists of estimating the relationships between variables. Like general supervised learning problems, a regression problem has a set of input variables or predictors and an output or dependent variable. In the case of GARUM, the predictors are the individual similarity measures, i.e., Sim_{hier}, $\text{Sim}_{\text{neigh}}$, $\text{Sim}_{\text{shared}}$ and Sim_{attr} (see Figure 4.15). The dependent variable is defined by a gold standard similarity measure, e.g., SeqSim in CESSM or a human-based similarity measure in the case of Lee50. Thus, a regression algorithm produces as output a function $f : X^n \rightarrow Y$, where X^n represents the predictors and Y corresponds to the dependent variable. Hence, GARUM can be defined as a function f:

$$\text{GARUM}(e_1, e_2) = f(\text{Sim}_{\text{hier}}, \text{Sim}_{\text{neigh}}, \text{Sim}_{\text{shared}}, \text{Sim}_{\text{attr}})$$

Depending on the type of regression, f can be a linear or a non-linear combination of the predictors. In both cases and regardless the used regression algorithm used, f is computed by minimizing a loss function. In the case of GARUM, the loss function is the mean squared error (MSE) which is defined as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2,$$

where Y is a vector of n observed values, i.e., gold standard values, and $\hat{Y}$ is a vector of n predictions, i.e., $\hat{Y}$ corresponds to results of the computed function f. Hence, the regression algorithm implemented in GARUM learns from a training dataset how to combine the individual similarity measures by means of a function f, such that the MSE among the results produced by f and the corresponding gold standard (SeqSim, ECC, etc.) is minimized.

Implementation

Given that GARUM utilizes supervised learning algorithms, training data is required. However, gold standards are usually defined for annotation sets, i.e., sets of knowledge graph resources, instead of for individual knowledge graph resources. CESSM, Lee50, or STS datasets are a good examples of this phenomenon, where real world

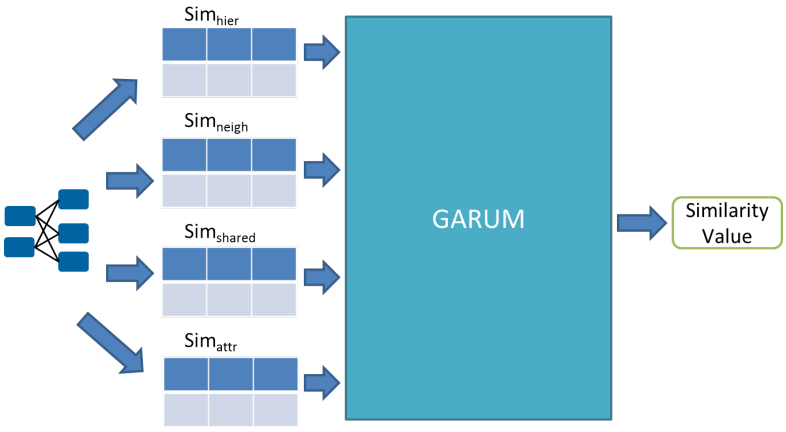


Figure 4.16: Workflow of the combination function for the individual similarity.

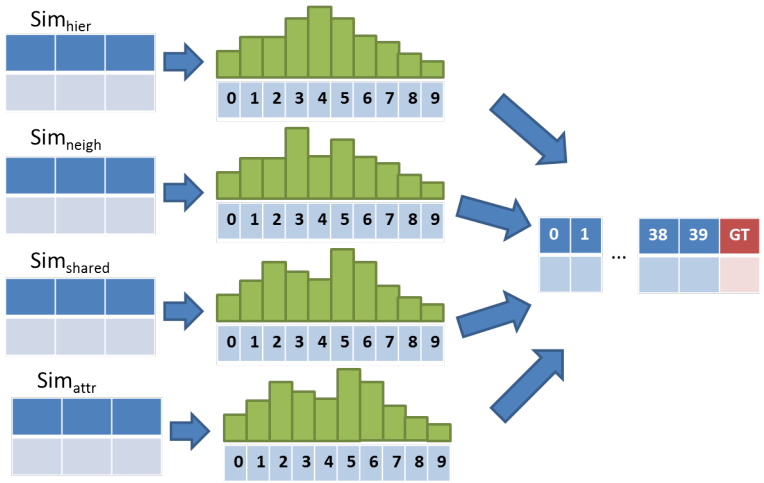


Figure 4.17: Transformation of the input matrices for the combination function.

entities (proteins or texts) are annotated with sets of knowledge graph resources. Gold standards provide values for such entities and not for the knowledge graph resources. Thus, the workflow of GARUM is represented in Figure 4.16. Instead of knowledge graph resources, the input consists of two sets of knowledge graph resources to be compared. Based on these sets, a similarity matrix for each considered individual similarity measure is computed. The output represents the aggregated similarity value computed by the estimated regression function f.

Classical machine learning algorithms have a fix number of input features. However, the dimensions of the matrices depend on the cardinality of the compared sets. Hence, the matrices cannot be directly used, but a transformation to a fixed structure is required. Figure 4.17 introduces the matrix transformation. For each matrix, a density histogram with 10 bins is created. Thus, the input dimensions are fixed to $10 \cdot |\text{Individual similarity measures}|$. In Figure 4.16, the input is an array with 40 features.

Finally, the transformed data is used to train the regression algorithm. This algorithm learns, based on the input, how to combine the value of the histograms to minimize the MSE with respect to the ground truth (GT in Fig. 4.17).

4.4.3 Evaluation

GARUM is evaluated on CESSM 2008, CESSM 2014, and Lee50. CESSM 2008 contains 13,430 comparisons, while CESMM reaches the 22,000 comparisons. Lee50 includes 1275 article comparisons. The STS dataset contains around 700 sentence comparisons. The annotation sets of these sentences are small (among one and three

annotations per sentence). The size of the annotation sets does not allow to apply supervised learning on them. For the rest of datasets three individual similarity measures are considered: Sim_{hier}, Sim_{neigh}, and Sim_{shared}. Hence, the input matrix has 30 features to be considered (10 per individual similarity measure). Neural networks and SVR [18] are used as supervised learning algorithms within GARUM and estimate the relevance of each individual measure.

Implementation During the evaluation of GARUM neural networks and SVR are used as supervised learning algorithms. Both algorithms are implemented in Python 2.7. The SVR version is implemented through the SVR class available in the the scikit-learn library[74]. The linear and the RBF kernels are evaluated. The neural network is implemented with the Keras[13] library for neural networks using TensorFlow [14] as backend. Figure 4.18 describes the structure of the configured neural network. The input layer of a neural networks must have so many neurons as input features. In the evaluation three individual similarity measures are considered and for each one a histogram with 10 bins is computed. Hence, the input layer has 30 neurons. The middle layer consists of 15 neurons, the half of the neurons in the input layers. This reduction put pressure on the network during the training phase to summarize the relevant information in the input data. Finally, the output neuron combines the 15 outputs of the middle layer to provide a similarity value.

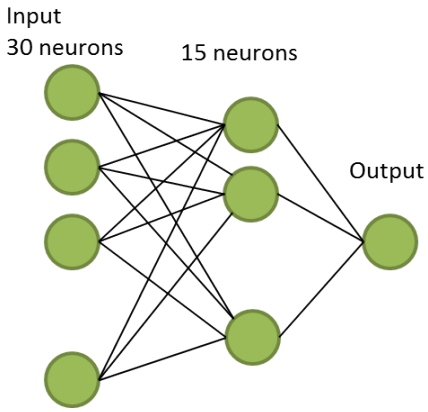


Figure 4.18: Configuration of the neural network used for GADES in CESSM 2008, CESSM 2014 and Lee50.

Results Table 4.15 contains the Pearson's correlation coefficient for the measures GADES and GARUM in CESSM 2008 and CESSM 2014. Individual similarity measures are combined in GADES through the product triangular norm. Then, the 1-1 maximum weighted matching is computed and the similarity values aggregated with the Hungarian Algorithm. GARUM uses a regression algorithm to combine the individual similarity measures and weight them according to their relevance. In this case, the regression algorithm decides both how the individual similarities are combined and how the different values are aggregated. The neural network is trained once per each gold standard measure and dataset. To ensure the quality and correctness of the evaluation, both datasets are split following a 10-cross fold validation strategy. Table 4.15 contains the Pearson's coefficient of GARUM for CESSM 2008 and 2014. The results represents the average correlation among the 10 folds. According to the results, the linear combination of the individual similarity measures (Linear SVR) is not effective, being non-linear combinations (RBF SVR and neural networks) able to obtain the best results for both datasets and therefore, to outperform GADES. GARUM could not be evaluated in CESSM 2014 with ECC and Pfam due to the lack of training data.Similarly, Table 4.16 shows that GARUM also outperforms the rest of similarity measure in the Lee50 dataset with a correlation value of 0.74 with the neural network shown in Figure 4.18.

[13] https://keras.io/
[14] https://www.tensorflow.org/

Similarity measure	2008			2014		
	SeqSim	*ECC*	*Pfam*	*SeqSim*	*ECC*	*Pfam*
GADES	0.773	0.437	0.546	0.836	0.515	0.49
GARUM Linear SVR	0.68	0.48	0.48	0.78	-	-
GARUM RBF SVR	**0.862**	**0.7**	**0.7**	0.864	-	-
GARUM NN	0.85	0.6	0.696	**0.878**	-	-

Table 4.15: Pearson's coefficient of GADES and GARUM for *SeqSim*, *ECC* and *Pfam* in CESSM 2008 and CESSM 2014.

Similarity measure	**Pearson's coefficient**
GADES	0.727
GARUM Linear SVR	0.726
GARUM RBF SVR	0.73
GARUM NN	**0.74**

Table 4.16: Lee50 results. Pearson's coefficient for GADES and GARUM in the Lee et al. knowledge graph [54].

4.5 Conclusions

In this chapter four semantic similarity measures are presented. On one hand, OnSim and IC-OnSim are specific for comparing ontology terms. OnSim considers the hierarchy and neighborhood as resource characteristics, while IC-OnSim considers additionally the shared information among the terms. Further, both measures use an OWL2 reasoner to infer knowledge in the ontology and obtain provenance information about the inferences, i.e., the justifications. Both measures outperform state-of-the-art similarity measures and exhibits good performance in CESSM 2008 and CESSM 2014. On the other hand, GADES is a *metric* suitable for any kind of graph data and takes into account the four identified resource characteristics: hierarchy, neighborhood, shared information and attributes. Though GADES does not use any OWL2 reasoner, it is able to be competitive with respect to OnSim and IC-OnSim in terms of accuracy in both versions of CESSM and performs better in terms of time. Thus, GADES can be also used in near real-time scenarios. Finally, GARUM, a similarity measure able to aggregate individual similarity measure by means of a machine learning algorithm, is presented. During the evaluation of GADES several aggregation strategies are used depending on the evaluation dataset. Hence, a machine learning approach based on regression algorithms is presented to overcome this problem and find the best way of combining the different individual similarity measures based on the considered resource characteristics. The results show that GARUM finds better combination ways than humans and outperforms GADES in CESSM and Lee50.

5 Semantic based Approaches for Enhancing Knowledge Graph Quality

Graphs offer a more flexible way of representing data without needing a fixed schema like relational databases. Nevertheless, graphs also suffer of quality problems like incompleteness, redundancy or obsolescence. In this chapter the benefits of considering semantics for enhancing knowledge graph quality are shown in three different tasks. First, a semantic-based approach to discover missing relations in knowledge graph is presented in Section 5.2. Incompleteness in knowledge graphs may cause approaches and tools relying on them, e.g. search engines, to fail and to deliver false or incomplete results. The automatic discovery of missing relations in knowledge graph can alleviate the effects of the incompleteness problem. Second, an approach to monitor the evolution of annotation graphs is introduced in Section 5.3. Monitoring the evolution of annotation graphs allows to better detect quality issues and anomalies that may affect the methods relying on such graphs. Finally, a semantic-based approach for data integration is described in Section 5.4. The same real world entity may be described in different knowledge graphs. Each knowledge graph can cover or describe different aspects of the entity. Thus, in order to obtain a full view of the information regarding the entity is necessary to query all of them. However, this is not possible if the different descriptions are not connected, i.e., integrated. An approach able to identify descriptions that potentially refer to the same real world entity can alleviate the effects of this quality problem.

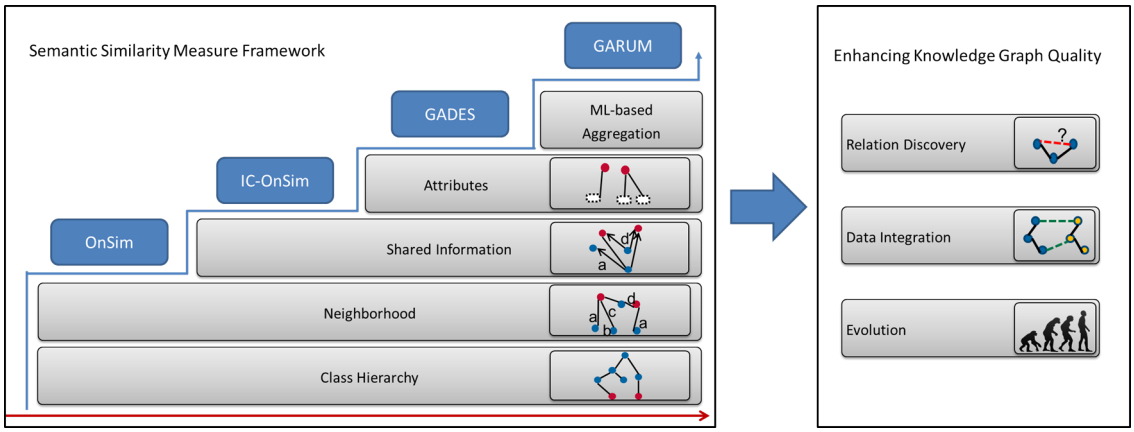


Figure 5.1: Set of data driven tasks for the enhancement of knowledge graph quality

5.1 Introduction

Semantic Web and Linked Data communities together with some of the big IT companies as Google [95, 31], Microsoft [31] or LinkedIn [36] have fostered during the last years the representation of data using graphs at the expense of relational databases. Graphs do not follow a fixed schema like relational databases and consider relationships as first-class citizen. Hence, they are more suitable to represent semantics and store sparse data. Nevertheless, the curation of graph based data is a tedious task and issues like incompleteness or inconsistency are still valid in graph based data [8, 73]. Thus, the vast amount of entities and relations in the graph makes graph curation a prone to errors task and humans may omit some attributes or relations between graph entities during the curation process. Further, unlike relational databases, which are used in local domains (a company, an institution,

etc.), Linked Data datasets aim to be public and open and a certain real world entity may be described in different datasets. Given that it is not expected to have a unique key for each real world entity, relating the distributed descriptions of real world entities is a new required task.

In this chapter, the benefits of considering semantics to enhance graph data quality are studied. Particularly, the chapter focuses on the incompleteness issue and the evolution. In Section 5.2 a semantic based relation discovery approach called KOI is presented. KOI considers the neighborhoods and the attributes as relevant resource characteristics to determine the similarity among knowledge graph entities. In Section 5.3 AnnEvol, an evolutionary framework for annotation graphs, is described. Finally, Section 5.4 introduces FuhSen, a similarity based integration approach provided with GADES as semantic-aware similarity measure.

5.2 Relation Discovery in Knowledge Graphs: KOI

In this section, KOI is presented, an approach for relation discovery in knowledge graphs that considers the semantics of both entities represented in the knowledge graph and their neighborhoods. KOI receives as input a knowledge graph, and encodes the semantics about the properties of graph entities and their neighbors in a bipartite graph. Entity neighbors correspond to ego-networks, e.g., the friends of a person in a social network or the set of TED talks related to a given TED talk. KOI partitions the bipartite graph into parts of highly similar entities connected to also similar neighborhood. Thus, the similarity is computed based on the attributes and the neighborhoods as relevant resource characteristics. Relations are discovered in these parts following the *homophily* prediction principle, which states that entities with similar characteristics tend to be related to similar entities [57]. Intuitively, the *homophily* prediction principle allows for relating two entities $t1$ and $t2$ whenever they have similar datatype and object property values, i.e., attributes and neighborhoods.

The behavior of KOI is evaluated on a knowledge graph of TED talks[1]; this knowledge graph was crafted by crawling data from the official TED website[2]. The relations discovered by KOI are compared with two baselines of relations identified by the METIS [43] and *k-Nearest Neighbors* (KNN) algorithms. KNN algorithm is empowered with statistic and semantic similarity measures (Section 5.2.4). Experimental outcomes suggest the following statements: *i*) Semantics encoded in similarity measures and knowledge graph structure enhances the effectiveness of relation discovery methods; and *ii*) KOI outperforms state-of-the-art approaches, obtaining higher values of precision and recall. To summarize, the contributions of this section are the following:

- KOI, a relation discovery method that implements graph partitioning techniques and relies on semantics encoded in similarity measures and graph structure to discover relations in knowledge graphs;
- A knowledge graph describing TED talks crafted from the TED website; and
- An empirical evaluation on a real world knowledge graph of TED talks to analyze the performance of KOI with respect to state-of-the-art approaches.

The work included in this section is published at the 20th International Conference on Knowledge Engineering and Knowledge Management [103].

5.2.1 Motivating Example

In this section, a motivating example of the tackled relation discovery problem is provided. Figure 5.2 shows a knowledge graph extracted from the TED website. Nodes represent TED talks, while edges between nodes indicate relatedness among the two connected talks. TED talks are described through textual properties, e.g., title, abstract or tags, and their relations with other talks in order to provide recommendations to the users. Relations

[1] https://github.com/itraveribon/TED_KnowledgeGraph
[2] http://www.ted.com/

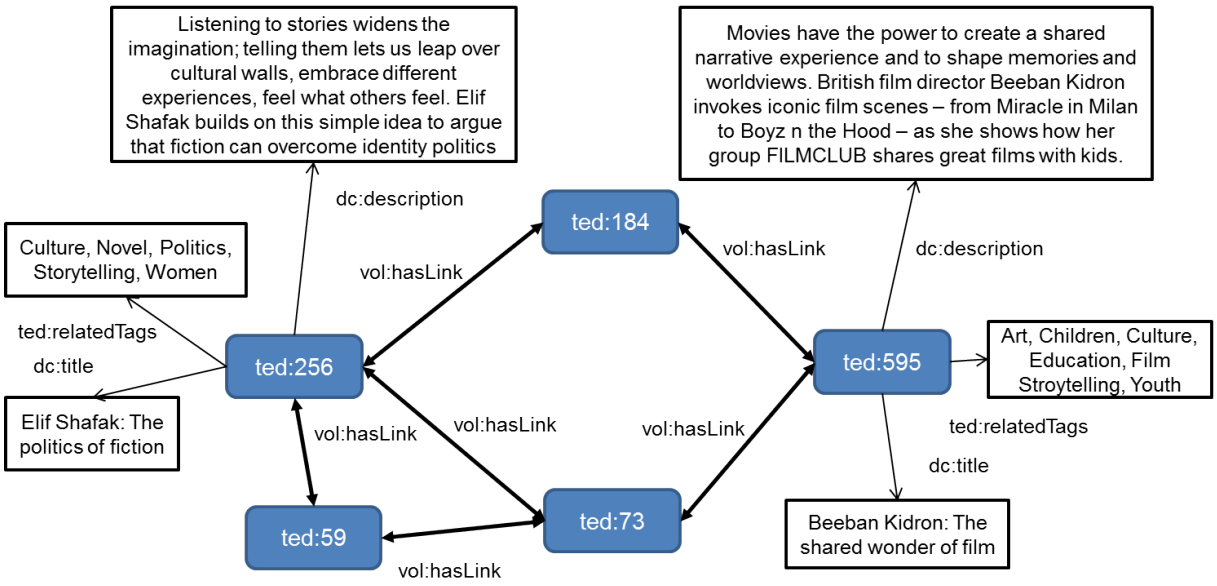


Figure 5.2: Portion of a knowledge graph of TED talks [103].

between talks are defined by TED curators manually, which constitutes a time expensive and prone to omissions task. Therefore, it would be helpful to have automatic methods able to ease the relation discovery tasks. In order to identify relations discovered by humans, two versions of the TED website (2015 and 2016) are compared. Then, the relations in 2016 that are not present in the 2015 version are marked as discoveries. In total, 62 relations are included in 2016 but are not present in the 2015 version, i.e., TED curators did not discover these relations until 2016. One example is the relation between talks *The politics of fiction*[3] and *The shared wonder of film*[4]. Both talks are present in both versions of the website. However, only in 2016 it is possible to find a relation between them. Thus, there are missing relations between TED talks in the 2015 version of the website, i.e., the knowledge graph is incomplete. An approach able to discover automatically these relations would alleviate the effort of curators and improve the quality (completeness) of the data.

Though the relation between *The politics of fiction* and *The shared wonder of film* is not included in the 2015 website, the rest of knowledge regarding to these talks allows for intuiting a high degree of relatedness between them. Both talks have keywords or tags in common as *Culture* or *Storytelling*. Some expressions can be found in their abstracts or descriptions, that though do not match exactly, are clearly related such as *identity politics* and *cultural walls*, or *film* and *novel*. Moreover, they share two related talks, *The clues to a great story*[5] and *The mistery box*[6] in their neighborhoods. Thus, these related talks have properties in common. KOI relies on this observation and exploits entity properties to discover missing relations between these entities. Particularly, KOI pays attention to two of the resource characteristics identified in the previous chapter: neighborhood and attributes.

5.2.2 Problem Definition

Let $G' = (V, E')$ and $G = (V, E)$ be two knowledge graphs. G' is an *ideal* knowledge graph that contains all the existing relations between entities in V. G is the *actual* knowledge graph, which contains only a portion of the relations represented in G', i.e., $E \subseteq E'$. Let $\Delta(E', E) = E' - E$ be the set of relations existing in the ideal graph that are not represented in the actual knowledge graph G, and $G_{\text{comp}} = (V, E_{\text{comp}})$ the *complete* knowledge graph, which contains a relation for each possible combination of entities and predicates $E \subseteq E' \subseteq E_{\text{comp}}$.

[3] http://www.ted.com/talks/elif_shafak_the_politics_of_fiction
[4] http://www.ted.com/talks/beeban_kidron_the_shared_wonder_of_film
[5] http://www.ted.com/talks/andrew_stanton_the_clues_to_a_great_story
[6] http://www.ted.com/talks/j_j_abrams_mystery_box

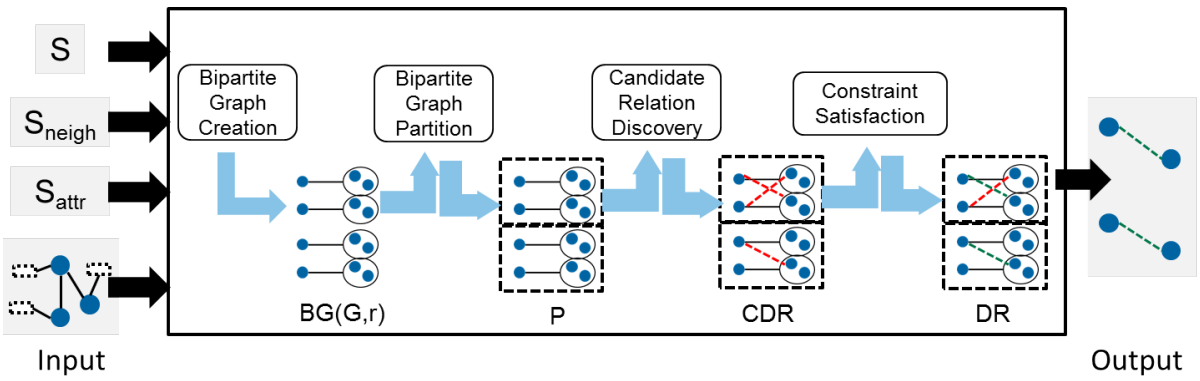


Figure 5.3: KOI Architecture [103].

Given a relation $e \in \Delta(E_{\text{comp}}, E)$, the *relation discovery problem* consists of determining if $e \in E'$, i.e., if a relation e corresponds to an existing relation in the ideal graph G'.

5.2.3 The KOI Approach

KOI is a relation discovery method for knowledge graphs that considers semantics encoded in similarity measures and the knowledge graph structure. KOI implements an unsupervised graph partitioning approach to identify parts of the graph from where relations are discovered. KOI applies the *homophily* prediction principle to each part of the partitioned bipartite graph, in a way that two entities with similar characteristics are related to similar entities. Similarity values are computed based on two resource characteristics: (a) the neighbors or ego-networks of the compared entities, and (a) their attributes or datatype property values (e.g., textual descriptions).

Figure 5.3 depicts the KOI architecture. KOI receives a knowledge graph $G = (V, E)$ like the one showed in Figure 5.2, two individual similarity measures Sim_{attr} and $\text{Sim}_{\text{neigh}}$, and a set of constraints S. As result, KOI returns a set of relations discovered in the input graph G. KOI builds a bipartite graph $BG(G, r)$ where each entity in V is connected with its ego-network according to the predicate r. Figure 5.4 contains the bipartite graph built from the knowledge graph in Figure 5.2 according to predicate *vol:hasLink*. By means of a graph partitioning algorithm and the similarity measures Sim_{attr} and $\text{Sim}_{\text{neigh}}$, KOI identifies graph parts containing highly similar entities with highly similar ego-networks, i.e., similar entities that are highly connected in the original graph (see P in Figure 5.3). According to the *homophily* prediction principle, KOI produces candidate missing relations inside the identified graph parts (see red edges in *CDR* in Figure 5.3) . Figure 5.5 represents with red dashed lines the set of candidate discovered relations. Only those relations that satisfy a set of constraints S are considered as discovered relations (see green edges *DR* in Figure 5.3). Listing 5.1 shows an example of constraints and Figure 5.6 includes the corresponding *score* values for each candidate relation.

Bipartite Graph Creation. Determining the membership of each relation $e \in \Delta(E_{\text{comp}}, E)$ in E' is expensive in terms of time due to the large amount of relations included in $\Delta(E_{\text{comp}}, E)$, and may produce a large amount of false positives. KOI leverages from the homophily intuition to tackle this problem by finding portions of the graph including entities with similar neighborhoods and similar attributes. In order to consider at the same time both similarities, KOI builds a bipartite graph where each entity is associated with its ego-network. The objective is to find a partitioning of this graph, such that each part contains highly similar entities and highly similar ego-networks. Thus, the KOI graph partitioning problem is an optimization problem where these two similarities are maximized on entities of each part. Figure 5.4 shows a KOI bipartite graph for the knowledge graph in Figure 5.2.

Definition 13. *KOI Bipartite Graph. Let $G = (V, E, L)$ be a knowledge graph, where L is a set of property labels. Given a predicate $r \in L$, the KOI Bipartite Graph of G and r is defined as $BG(G, r) = (V \cup U(r), E_{BG}(r))$, where*

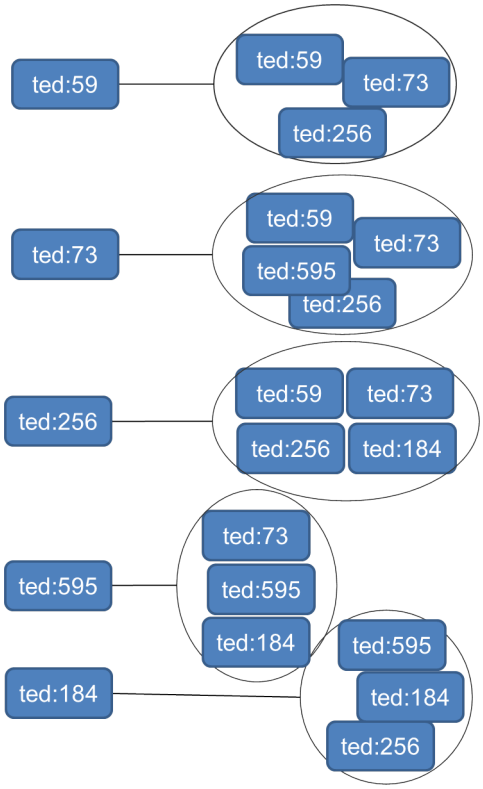


Figure 5.4: KOI Bipartite graph generated from Figure 5.2 [103].

$U(r) = \{ego\text{-}net(v_i, r) \mid v_i \in V\}$ is the set of ego-networks of entities in V, and $E_{BG}(r) = \{(v_i, u_i) \mid v_i \in V \wedge u_i = ego\text{-}net(v_i, r)\}$ is the set of edges that associate each entity with its ego-network.

Bipartite Graph Partitioning. To identify portions of the knowledge graph where the *homophily* prediction principle can be applied, the bipartite graph $BG(G, r)$ is partitioned in a way that entities in each part are highly similar (i.e., similar attributes) and connected (i.e., have similar ego-networks).

Definition 14. *A Partition of a KOI Bipartite Graph. Given a KOI bipartite graph $BG(G, r) = (V \cup U, E_{BG})$, a partition $P(E_{BG}) = \{p_1, p_2, ..., p_n\}$ satisfies the following conditions:*

- *Each part p_i contains a set of edges $p_i = \{(v_x, u_x) \in E_{BG}\}$,*
- *Each edge (v_x, u_x) in E_{BG} belongs to one and only one part p of $P(E_{BG})$, i.e., $\forall p_i, p_j \in P(E_{BG}), p_i \cap p_j = \emptyset$ and $E_{BG} = \bigcup_{p \in P(E_{BG})} p$.*

Definition 15. *The Problem of KOI Bipartite Graph Partitioning. Given a KOI bipartite graph $BG(G, r) = (V \cup U, E_{BG})$, and similarity measures Sim_{attr} and Sim_{neigh} for entities in V and ego-networks in U. The problem of KOI Bipartite Graph Partitioning corresponds to the problem of finding a partition $P(E_{BG})$ such that Density($P(E_{BG})$) is maximized, where Density($P(E_{BG})$)=$\sum_{p \in P(E_{BG})}$(partDensity(p)), and:*

- *Density($P(E_{BG})$)=$\sum_{p \in P(E_{BG})}$(partDensity(p)), and*
- $$partDensity(p) = \frac{\overbrace{\sum_{v_i, v_j \in V_p} [v_i \neq v_j] Sim_{attr}(v_i, v_j)}^{(A)}}{|V_p|(|V_p| - 1)} + \frac{\overbrace{\sum_{u_i, u_j \in U_p} [u_i \neq u_j] Sim_{neigh}(u_i, u_j)}^{(B)}}{|U_p|(|U_p| - 1)}$$

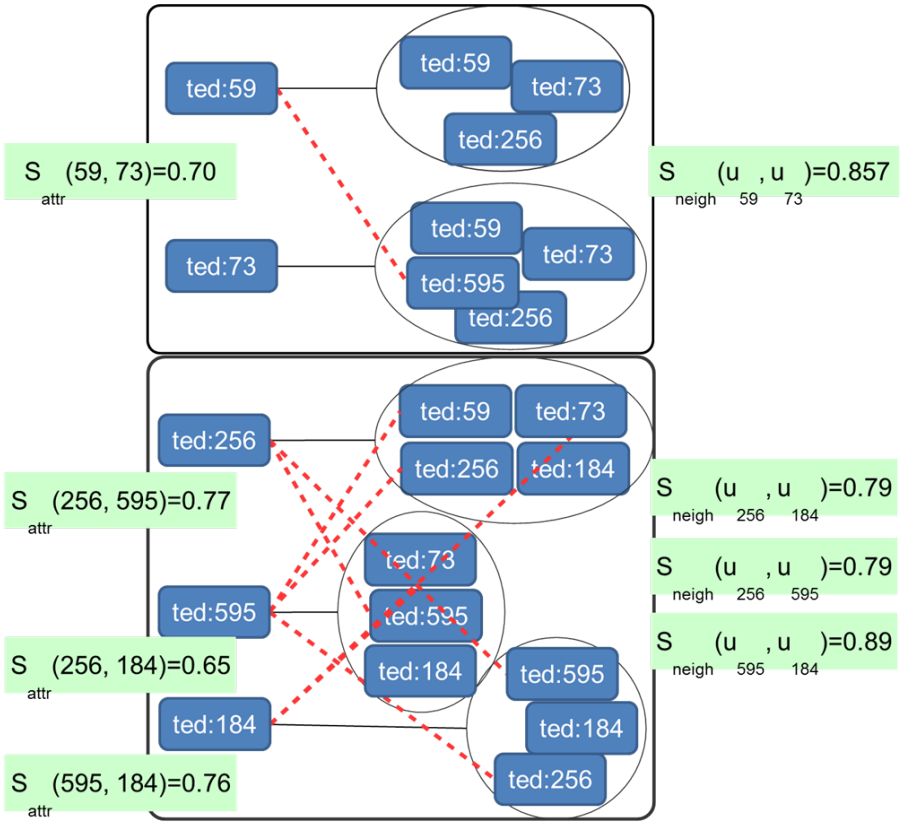


Figure 5.5: Partition found by the KOI Bipartite Graph in Figure 5.4 [103].

where component (A) represents the similarity between entities in edges of part p and (B) represents the similarity between the corresponding ego-networks. Sim_{attr} and Sim_{neigh} are similarity measures for attributes and ego-networks (neighborhoods), respectively.

KOI utilizes the partitioning algorithm proposed by Palma et al. [71] semEP to solve the optimization problem of partitioning a KOI bipartite graph. The bipartite graph in Figure 5.4 is partitioned into two parts represented in Figure 5.5. Squares represent partitions and red dashed edges candidate relations of our approach. Sim_{attr} and $\text{Sim}_{\text{neigh}}$ are entity and ego-network similarity measures, respectively. Entities of the part in the bottom are $V_p = \{ted{:}256, ted{:}595, ted{:}184\}$ and their corresponding ego-networks are $U_p = \{u_{256}, u_{595}, u_{184}\}$. In order to calculate the *partDensity* of this part, entities in V_p with Sim_{attr} and ego-networks in U_p with $\text{Sim}_{\text{neigh}}$ are pairwise compared. Thus, the similarity Sim_{attr} is computed for entity pairs $\text{Sim}_{\text{attr}}(ted{:}256, ted{:}595)$, $\text{Sim}_{\text{attr}}(ted{:}256, ted{:}184)$, and $\text{Sim}_{\text{attr}}(ted{:}595, ted{:}184)$, and the similarity $\text{Sim}_{\text{neigh}}$ for ego-networks pairs $\text{Sim}_{\text{neigh}}(u_{256}, u_{595})$, $\text{Sim}_{\text{neigh}}(u_{256}, u_{184})$, and $\text{Sim}_{\text{neigh}}(u_{595}, u_{184})$. The computed *partDensity* value is in this case 0.775.

Candidate Relation Discovery. KOI applies the *homophily* prediction principle in the parts of a partition of a KOI bipartite graph, and discovers relations between entities included in the same part.

Definition 16. *Candidate relation. Given two knowledge graphs $G = (V, E)$ and $G_{comp} = (V, E_{comp})$. Let $BG(G, r) = (V \cup U, E_{BG})$ be a KOI bipartite graph. Let $P(E_{BG})$ be a partition of E_{BG}. Given a part $p = \{(v_x, u_x) \in E_{BG}\} \in P(E_{BG})$, the set of candidate relations $CDR(p)$ in part p corresponds to the set of relations $\{(v_i, r, v_j) \in E_{comp}\}$ such that v_j is included in some ego-network u_x and edges $(v_i, u_i), (v_x, u_x) \in p$.*

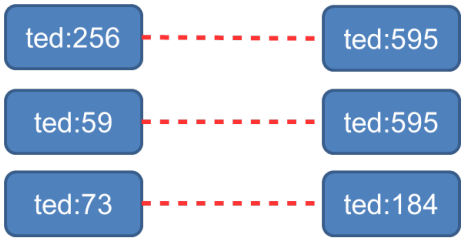


Figure 5.6: Application of the relation constraint described in Listing 5.1 for the candidate relations (red dashed edges) found in Figure 5.5.

In Figure 5.5 candidate relations are represented as red dashed lines. One example is the relation (*ted:59*, *vol:hasLink*, *ted:595*). This candidate relation is discovered due to the presence of *ted:59* and *ego-net(ted:73, vol:hasLink)* in the same part and the inclusion of the entity *ted:595* in the ego-network *ego-net(ted:73, vol:hasLink)*.

Constraint satisfaction. A relation constraint is a set of RDF constraints that states conditions that must be satisfied by a candidate discovered relation in order to become a discovered relation. RDF constraints are expressed using the SPARQL language as suggested by Lausen et al. [52] and Fischer et al. [23]. Only the candidate relations that fulfill relation constraints are considered as discovered relations.

Definition 17. *Discovered Relations. Given a set of candidate relations CDR and a set of relation constraints S, the set of discovered relations DR is defined as the subset of candidate relations that satisfy the given contraints* $DR(CDR,S) = \{(v_i,r,v_j) \mid (v_i,r,v_j) \in CDR \wedge \forall rc \in S \Rightarrow satisfy(rc(v_i,r,v_j))\}$.

The upper bound for checking if a constraint is satisfied by a candidate discovered relation is PSPACE-complete [76]. Nevertheless, given that the amount of constraints is lower than the size of the knowledge graph, the complexity of this problem can be expressed in terms of data, and is LOGSPACE for SPARQL [52, 76].

```
ASK  {
  { SELECT count(?uk) as ?repetitions
       count(?vunion) as ?union
       count(?vinter) as ?intersection
       ?intersection/?union
          * ?repetitions as ?score
       WHERE {
    vj dc:isPartOf ?uk.
    ?vk dc:relation ?uk.
    vi dc:relation ?ui.
    ?ui dc:isPartOf ?p.
    ?uk dc:isPartOf ?p.
    {?vunion dc:isPartOf ?uk} UNION
           {?vunion dc:isPartOf ?ui}.
    ?vinter dc:isPartOf ?uk.
    ?vinter dc:isPartOf ?ui.
    }
}
  FILTER (?score > THETA)
}
```

Listing 5.1: SPARQL specification of a constraint on the product between the similarity of the ego-networks and the amount of times a relation is discovered.

Listing 5.1 illustrates a constraint that states a condition for a candidate discovered relation $cdr = (v_i\ r\ v_j)$ to become a discovered relation. Whenever the candidate discovered relation $cdr = (v_i\ r\ v_j)$ is identified in several parts of a partition P, the number of times that cdr appears is taken into account, as well as the similarity between the ego-network of v_i and the ego-networks where v_j is included. To determine if the constraint is satisfied, a *score* is computed and the value of this score has to be greater than a threshold θ_i. The score is defined as the product

of the number of times a relation is discovered and the similarity between corresponding ego-networks. For each discovered relation, Figure 5.6 contains the value of the corresponding score described in Listing 5.1. Relation (*ted:256*, *vol:hasLink*, *ted:595*) gets the highest value for this score being discovered four times in Figure 5.5. Moreover, the similarity between ego-networks *ego-net*(*ted:595*,*vol:hasLink*) and *ego-net*(*ted:184*, *vol:hasLink*) is 0.5. The constraint, specified as an ASK query, is held if at least one score value is greater than the threshold θ. Therefore, only the maximum similarity value between the ego-networks is considered.

5.2.4 Empirical Evaluation

This section is divided in four parts. First, the creation of the knowledge graph used during the evaluation is explained and the properties included in the graph are detailed. Second, details about the implementation are given to ensure reproducibility and inform the reader about the metrics used to measure the quality of the approach. Following, semantic and no semantic similarity measures are used with KNN to discover relations in the described knowledge graph. Finally, the methods KOI and METIS [43] are evaluated in the same knowledge graph.

Knowledge Graph Creation

In this section the characteristics of the crafted TED knowledge graph and its links to external vocabularies are described. This knowledge graph is built from a real world dataset of TED talks and playlists[7].

The knowledge graph of TED talks consists of 846 talks and 125 playlists (15/12/2015). Playlists are described with a title and the set of included TED talks. Each TED talk is described with the following set of datatype properties or attributes:

- *dc:title* (Dublin Core vocabulary) represents title of the talk;
- *dc:creator* models speaker;
- *dc:description* represents abstract;
- and *ted:relatedTags* corresponds to set of related keywords.

Apart from the attributes, TED talks are connected to playlists that include them through the relation *ted:playlist*. A *vol:hasLink* (Vocabulary Of Links[8]) relation connects each pair of talks that share at least one playlist. Available playlists were crawled from the TED website[9]. Playlists contain sets of TED talks that usually address similar topics. TED curators create and maintain the playlists, deciding whether to include a certain TED talk playlist.

Additionally, the knowledge graph was enriched by adding similarity values between each pair of entities. Four similarity measures (TFIDF, ESA [26], Doc2Vec [53], and Doc2Vec Neighbors) were used as Sim_{attr}, receiving as input the concatenation of attributes title, description and related tags. ESA similarity values were computed using the public ESA endpoint[10], and Doc2Vec (D2V) values were obtained training the *gensim* implementation [81] with the pre-trained Google News dataset[11]. Doc2Vec Neighbors (D2VN) is defined as:

$$\text{D2VN}(v_1, v_2) = \frac{\sqrt{\text{Sim}_{\text{attr}}(v_1, v_2)^2 + \text{Sim}_{\text{neigh}}(\text{ego-net}(v_1, r), \text{ego-net}(v_2, r))^2}}{\sqrt{2}},$$

where *r* corresponds to *vol:hasLink* and Sim_{attr} and $\text{Sim}_{\text{neigh}}$ are defined as follows:

$$\text{Sim}_{\text{attr}}(v_1, v_j) = Doc2Vec(v_i, v, j)$$

[7] Data collected on 15/2/2015 and 22/04/2016
[8] `http://purl.org/vol/ns/`
[9] `http://www.ted.com/playlists`
[10] `http://vmdeb20.deri.ie:8890/esaservice`
[11] Google pre-trained dataset: `https://goo.gl/flpokK`

$$\text{Sim}_{\text{neigh}}(V_1, V_2) = \frac{2 * \sum_{(v_i, v_j) \in WEr} \text{Doc2Vec}(v_i, v_j)}{|V_1| + |V_2|}$$

where WEr represents the set of edges included in the 1-1 maximal bipartite graph matching following the definition of Schwartz et al. [90].

Unlike the knowledge graph created by Taibi et al. [98], our knowledge graph of TED talks includes information about the playlists, the relations between TED talks, and four similarity values for each pair of talks (TFIDF, ESA, Doc2Vec, and Doc2Vec Neighbors). The knowledge graph of TED talks is available at `https://goo.gl/7TnsqZ`.

Experimental configuration

The effectiveness of KOI to discover missing relations is empirically evaluated in the 2015 TED knowledge graph, which is based on a real world dataset. KOI is compared with METIS [43] and *k-Nearest Neighbors* (KNN) empowered with four similarity measures: TFIDF, ESA, Doc2Vec, and Doc2Vec Neighbors.

Research Questions: This empirical study aims at answering the following research questions: **RQ1**) Does semantics encoded in similarity measures affect the relation discovery task? In order to answer this question four similarity measures are compared, one statistical-based measure (TFIDF) and three semantic similarity measures (ESA [26], Doc2Vec [53], and Doc2Vec Neighbors). Doc2Vec Neighbors considers both, the semantics encoded in attributes and the neighborhood by taking into account the ego-networks. **RQ2**) Is KOI able to outperform common discovery approaches like METIS or KNN?

Implementation: KOI was implemented in Java 1.8 and the experiments were executed on an Ubuntu 14.04 64 bits machine with CPU: Intel(R) Core(TM) i5-4300U 1.9 GHz (4 physical cores) and 8GB RAM. In order to perform a fair evaluation, 10-fold cross-validation strategy was used to split the dataset into train and test datasets. For this splitting the library *WEKA* [32] version 3.7.12 was used. The cross-validation was performed over the set of relations among TED talks. In order to discover relations with METIS, the METIS solver version 5.1[12] was used. METIS receives as input a KOI Bipartite Graph with the same similarity measures Sim_{attr} and $\text{Sim}_{\text{neigh}}$ above specified for KOI. METIS returns a partitioning of the given graph, and candidate discovered relations are produced as explained in Section 5.2.3. In order to perform a fair comparison, the same constraint (Listing 5.1) is applied for the results of both, KOI and METIS.

Evaluation metrics: For each discovery approach, the following metrics are computed: i) *Precision*: Relation between the number of correctly discovered relations and the whole set of discovered relations. ii) *Recall*: Relation between the number of correctly discovered relations and the number of existing relations in the dataset. iii) *F-Measure*: harmonic mean of precision and recall. Values showed in Tables 5.1 and 5.2 are the average values over the 10-folds. Moreover, the F-Measure curves for KOI and METIS are drawn and the Precision-Recall Area Under the Curve (AUC) coefficients are calculated (Table 5.3).

Discovering relations with K-Nearest Neighbors

In the first experiment, relations are discovered in the graph using the K-Nearest Neighbors (KNN) algorithm under the hypothesis that highly similar TED talks should be related. Given a talk, relations between it and its K most similar talks are discovered. This experiment evaluates the impact of considering semantics encoded in domain similarity measures during the relation discovery task (**RQ1**).

Table 5.1 reports on the results obtained by four similarity measures: TFIDF, ESA [26], Doc2Vec [53], and Doc2Vec Neighbors. The first three similarity measures only consider knowledge encoded in attributes. On the

[12] `http://glaros.dtc.umn.edu/gkhome/metis/metis/download`

K	Precision				Recall				F-Measure			
	TFIDF	ESA	D2V	D2VN	TFIDF	ESA	D2V	D2VN	TFIDF	ESA	D2V	D2VN
2	0.219	0.251	0.300	0.558	0.036	0.042	0.048	0.156	0.06	0.07	0.08	0.244
3	0.203	0.240	0.286	0.424	0.050	0.060	0.069	0.212	0.077	0.092	0.107	0.283
4	0.191	0.220	0.267	0.322	0.061	0.072	0.085	0.257	0.089	0.104	0.123	0.285
5	0.179	0.205	0.255	0.254	0.072	0.083	0.101	0.288	0.098	0.113	0.138	0.27
6	0.172	0.196	0.243	0.208	0.083	0.094	0.115	0.322	0.106	0.121	0.148	0.253
7	0.165	0.187	0.235	0.175	0.092	0.104	0.129	0.35	0.112	0.127	0.158	0.233
8	0.158	0.177	0.227	0.149	0.101	0.111	0.142	0.373	0.117	0.129	0.165	0.233
9	0.153	0.169	0.217	0.128	0.110	0.120	0.152	0.391	0.12	0.132	0.168	0.193
10	0.147	0.160	0.212	0.113	0.118	0.126	0.165	0.41	0.123	0.133	0.175	0.177
11	0.143	0.154	0.207	0.1	0.124	0.133	0.177	0.422	0.133	0.135	0.18	0.171
12	0.139	0.149	0.200	0.089	0.132	0.140	0.186	0.434	0.128	0.137	0.181	0.147
13	0.134	0.144	0.195	0.08	0.138	0.146	0.184	0.442	0.128	0.136	0.196	0.135
14	0.131	0.138	0.190	0.072	0.145	0.151	0.205	0.45	0.129	0.136	0.186	0.125

Table 5.1: Effectiveness of KNN. D2V = Doc2Vec, D2VN = Doc2Vec Neighbors.

other hand, Doc2Vec neighbors compares two entities considering the knowledge located in attributes and the structure of the graph by taking into account the ego-networks. Results obtained with the first three similarity measures suggest that Doc2Vec and ESA, which are semantic similarity measures, are able to outperform TFIDF, which does not take into account semantics. Doc2Vec obtains the highest F-measure value (0.196) with $K = 13$, which is significantly better than the maximum values obtained by ESA (0.137) and TFIDF (0.133). Thus, results allow to conclude that considering semantics encoded in Doc2Vec has a positive impact in the relation discovery task with respect to ESA and TFIDF. Results obtained with the Doc2Vec Neighbors indicate that knowledge encoded in ego-networks is of great value and that combining it with the knowledge encoded in attributes allows for obtaining a higher F-measure value (0.285) than the other three similarity measures.

Effectiveness of KOI discovering relations

KOI was executed using the definitions of Sim_{attr} and $\text{Sim}_{\text{neigh}}$ provided for D2VN in Section 5.2.4. In this section KOI is compared with respect to METIS [43] using the relation constraint constraint defined in Listing 5.1. Sim_{attr} and $\text{Sim}_{\text{neigh}}$ are defined as follows:

$$\text{Sim}_{\text{attr}}(v_1, v_j) = Doc2Vec(v_i, v, j)$$

$$\text{Sim}_{\text{neigh}}(V_1, V_2) = \frac{2 * \sum_{(v_i, v_j) \in WEr} \text{Doc2Vec}(v_i, v_j)}{|V_1| + |V_2|}$$

Table 5.2 summarizes the results of KOI and METIS. Values of θ correspond to the value of variable THETA of the constraint in Listing 5.1. The highest F-measure value is 0.512 and is obtained by KOI with $\theta = 0.7$. This F-measure value is higher than the one obtained with KNN and Doc2Vec Neighbors (0.285) and also higher than the maximum value obtained by METIS (0.39). The parameter θ can be configured depending on the respective importance of precision and recall. Lower values of θ deliver high values of recall, while high values of θ deliver high values of precision. Figure 5.7 shows the F-Measure curve for values of $\theta \in [0, 2]$. Area under the curves indicates quality of the approaches. KOI is able to get higher F-Measure values for almost all θ values. Further, the Precision-Recall Curve for KOI, METIS and KNN Doc2Vec Neighbors is computed. Table 5.3 shows that KOI gets a higher AUC value (0.396) than METIS (0.244) and KNN (0.223). Thus, KOI outperforms both, METIS and

θ	KOI			METIS		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
0	0.068	0.645	0.123	0.011	0.732	0.022
0.7	0.563	0.47	0.512	0.218	0.553	0.31
0.8	0.633	0.424	0.507	0.234	0.518	0.319
0.9	0.646	0.374	0.473	0.251	0.478	0.326
1	0.678	0.356	0.466	0.271	0.458	0.337
1.1	0.728	0.343	0.465	0.305	0.439	0.356
1.2	0.776	0.334	0.466	0.336	0.428	0.373
1.3	0.808	0.321	0.46	0.367	0.413	0.385
1.4	0.853	0.304	0.448	0.392	0.393	0.389
1.5	0.867	0.287	0.431	0.41	0.378	0.39
1.6	0.887	0.265	0.408	0.432	0.357	0.388

Table 5.2: Comparison of KOI and METIS.

Approach	AUC	F-Measure
KOI	**0.396**	**0.512**
METIS	0.244	0.39
KNN D2VN	0.223	0.285

Table 5.3: Area Under the Curve coefficients for KOI, KNN Doc2Vec Neighbors and METIS.

KNN Doc2Vec Neighbors, discovering more relations than KNN Doc2Vec Neigbors (higher recall) and a higher proportion of correct relations than METIS (higher precision for the same recall).

5.3 Evolution of Annotation Datasets: AnnEvol

Semantic Web initiatives have promoted the collaborative definition of ontologies which have been used to semantically describe and annotate entities from different domains. Particularly, the Biomedical Science has greatly benefited from these research movements, and expressive ontologies have been defined, e.g., the Gene Ontology (GO)[13] and the Human Phenotype Ontology (HPO)[14]. Ontologies in the biomedical domain have been extensively accepted by the scientific community as standards to describe concepts and relations, and to replace textual descriptions by controlled vocabulary terms from the ontologies.

Annotation quality can be impacted by the evolution of the ontology, changes in the annotations and type of annotation. Ontology terms can be incorporated or eliminated from the ontologies, as well as annotations that describe scientific entities. Additionally, the evolution of the ontology and the annotations is not always monotonic, i.e., ontology terms and annotations can be removed from the ontology or the annotation set. Further, not all the annotated entities are equally studied and therefore stability of their annotations is non-uniform. For example, if the scientific community is focused on a certain protein or disease, the annotations of this protein or disease will change more frequently than the description of other entities. In order to describe the quality of the annotations, Gross et al. [30] propose an *evolution model* able to represent different types of changes in ontology-based annotated entities, quality of the changed annotations, and the impact of the ontology changes. In this direction, Skunca et al. [94] define three measures to compute the annotation quality of computationally predicted GO annotations.

[13] `http://geneontology.org/`
[14] `http://www.human-phenotype-ontology.org/`

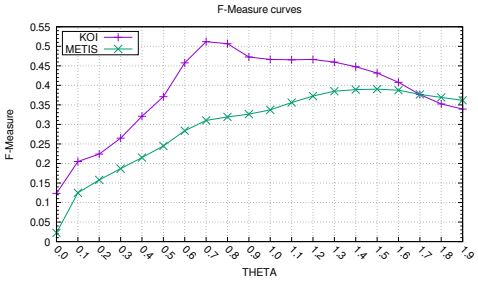


Figure 5.7: F-Measure curves of KOI and METIS [103].

Both approaches are able to describe datasets of ontology-based annotated entities in terms of the evolution of the annotations over time, i.e., both approaches provide an *annotation-wise description* of ontology-based annotated entities. Conducting an *annotation-wise description* of the evolution of annotations allows for the discovery of relevant patterns, e.g., stability of the GO annotations of Swiss-Prot versus Ensembl annotations [30] or improvement of GO computationally inferred annotations. Nevertheless, changes of groups of annotations into similar annotations, elimination of groups of obsolete annotations, as well as the inclusion of recently annotations cannot be expressed, i.e., an *annotation set-wise description* of the ontology-based annotated entities is not performed.

In this section *AnnEvol*, an evolutionary framework able to perform an *annotation set-wise description* of the evolution of an ontological annotated dataset, is presented. *AnnEvol* compares two sets of annotations d_i and d_{i+1} described within an ontology O in terms of the following parameters: *i*) *group evolution* captures how groups of annotations in d_i evolve into groups of similar annotations in d_{i+1}, semantic similarity measures are used to compute similarity of annotations; *ii*) *unfit annotations* measures the number of annotations that are used in d_i but do not survive in d_{i+1}; *iii*) *new annotations* measures the number of annotations that are used in d_{i+1} but are not in d_i; *iv*) *obsolete annotations* measures the number of annotations that are used in d_i but did not survive in d_{i+1} because they became obsolete in O; and *v*) *novel annotations* measures the number of annotations that are not used in d_i but are used in d_{i+1} after being included as part of O. The expressive power of *AnnEvol* is studied on four versions of the set of proteins available at the online tool Collaborative Evaluation of Semantic Similarity Measures (CESSM)[15]. The reported results suggest that annotations gradually evolve into groups of similar annotations, as well as that the evolution of the annotations not only depends on the type of organism of the protein (e.g., Homo Sapiens), but also to the type source of the annotation, i.e., Swiss-Prot and UniProt-GOA.

AnnEvol is published at the 11th International Conference on Data Integration in the Life Sciences [106].

5.3.1 Motivating Example

The benchmark CESSM 2008 is evaluated over 1,033 annotated proteins. The annotations of these proteins evolve over the time. Hence, the annotation sets in February 2008, December 2010, November 2012, and November 2014 are downloaded from the UniProt-GOA dataset [16]. *AnnSim* [70] is used to compare consecutive annotation sets of each protein. *AnnSim* performs a 1-1 perfect matching among the annotation sets. Edges in the corresponding bipartite graph are weighted with D_{tax}.

The hypothesis is that the evolution of annotations enhances the knowledge about the proteins and increases the correlation coefficients among *AnnSim* and the gold standard similarity measures *ECC* [17], *Pfam* [79], and *SeqSim* [78]. Table in Table 5.4 reports on the Pearson's coefficient between *AnnSim*, and *ECC*, *Pfam* and *SeqSim*. Contrary to the initial hypothesis, the correlation values do not improve over time. This may indicate that the quality of the annotations either does not improve over the time or the annotation sets evolve ununiformly.

[15] http://xldb.di.fc.ul.pt/tools/cessm/about.php
[16] http://www.uniprot.org/

	2008	2010	2012	2014
SeqSim	**0.65**	0.61	0.56	**0.56**
ECC	**0.39**	0.38	0.38	**0.38**
Pfam	**0.46**	0.45	0.43	**0.43**

Table 5.4: Pearson's coefficient between *AnnSim*, and *ECC*, *Pfam* and *SeqSim* for four annotation versions of UniProt-GOA proteins in CESSM 2008

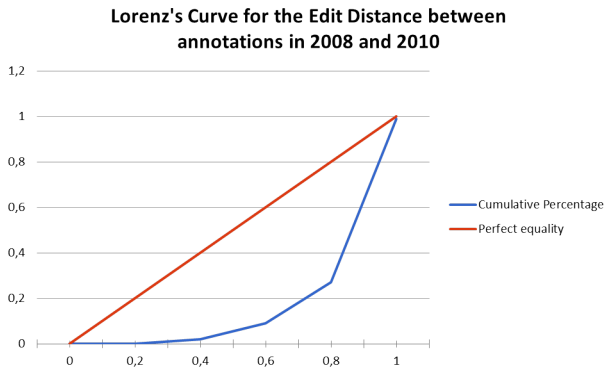


Figure 5.8: Lorenz's curve of the reflexive edit distances of 2008-2010

A 1-1 maximum weight bipartite match is performed between the annotation sets of the compared proteins. This kind of matching assigns low similarity values to pairs of proteins with very different number of annotations. The pair of proteins `P48734` and `P06493` justify clearly the worsening of the 1-1 perfect matching values over this time period. SeqSim returns a high similarity value of 0.99 for `P48734` and `P06493`. However, the 1-1 perfect matching returns 0.81, 0.24, 0.23, and 0.21 in the datasets of 2008, 2010, 2012, and 2014, respectively. This phenomenon is due to the non-uniform evolution of the annotation of these proteins. Annotations of `P06493` increase from 7 annotations in 2008 to 55 in 2010, 70 in 2012, and 76 in 2014. Further, protein `P48734` changes from having 6 annotations in 2008 to 8 in 2010, 9 in 2012, and 10 in 2014. In order to measure the uniformity, the Gini's coefficient of the edit distance between the annotations of each protein in UniProt-GOA is computed for generation changes 2008-2010, 2010-2012, and 2012-2014. Three operations are considered in the annotation set: annotation addition, annotation removing, and annotation substitution. The Gini's coefficient returns values between 0.0 and 1.0, where values close to 0.0 indicate, in this case, perfect equality in the evolution of the annotations, while values close to 1.0 indicate maximal inequality. The Gini's coefficient returns the values 0.65, 0.58, and 0.63 for the respective transitions. Figure 5.8 presents the Lorenz's curve for the transition 2008-2010. The horizontal axis represents the cumulative percentage of comparisons and the vertical axis the edit distance values. The red line indicates how would be a perfect equal evolution in terms of the defined edit distance. The blue line corresponds to the edit distance distribution for this transition. Hence, the number of changes in the annotations of the resources are non-uniformly distributed. The number of annotation per protein is also non-uniform. Gini's coefficients for the annotation distribution for each dataset version are 0.40, 0.44, 0.45, and 0.45, respectively. Figure 5.9 shows the Lorenz's curve for annotations in the version of 2014. The horizontal axis represents the cumulative percentage of proteins and the vertical axis corresponds to the percentage of annotations. The increase of the inequality of the distribution of annotations per protein and the inequality of the evolution of the proteins justify the worsening of the 1-1 perfect matching as aggregation function. Therefore, *AnnSim* or any other similarity measure, like GADES, whose values are aggregated with a 1-1 perfect matching will not improve their accuracy due to the not uniform evolution. Thus, a framework to monitor and analyze the evolution of annotation sets is needed.

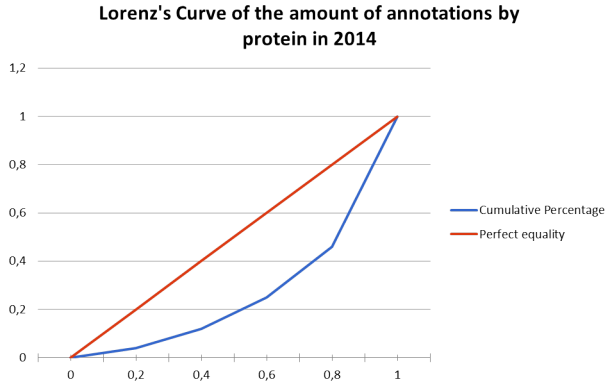


Figure 5.9: Lorenz's curve for distribution of annotation per protein in the version of 2014 of the annotation dataset

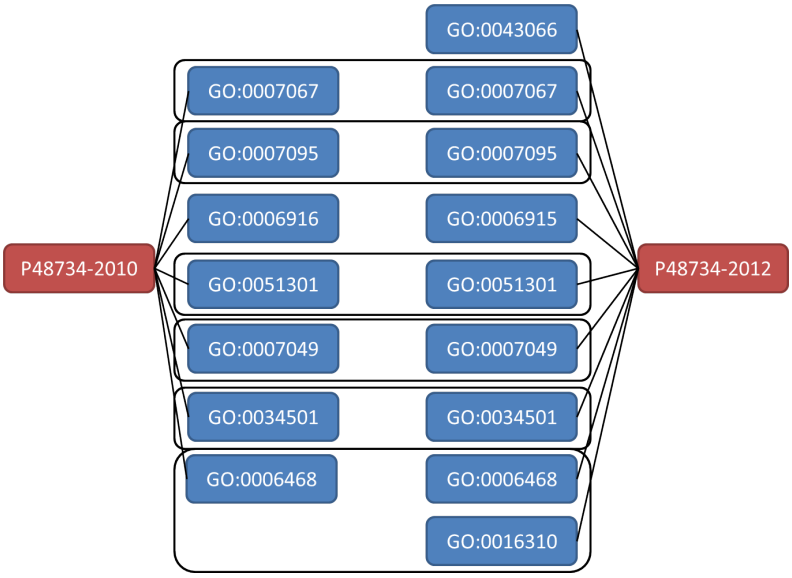


Figure 5.10: Clusters (black rectangles) formed by semEP between annotations of protein P48734 in 2010 and 2012.

5.3.2 Evolution of Ontology-based Annotated Entities

Annotations associated with entities may change over time, e.g., protein P48734 in Figure 5.10 suffers several changes from 2010 to 2012: *i*) the annotation GO:0006916 is removed, and *ii*) the annotations GO:0016310, GO:0006915, and GO:00043066 are added. A generation of an entity e, e.g., a protein, at time i is defined as a pair $g(e,i) = (A(e,i), O_i)$, where $A(e,i)$ is the annotation set of entity e at time i and O_i is the ontology version at time i. Then, to measure the evolution between two generations of the annotations of an ontology-based annotated entity e, the sets of annotations A_i and A_{i+1} are compared, e.g., the annotations of the protein P48734 at 2010 and 2012. The generation change of an annotated entity between the generations $g(e,i) = (A(e,i), O_i)$ and $g(e,i+1) = (A(e,i+1), O_{i+1})$ is defined as a quintuple $q(e|i,i+1) = (s,n,w,o,v)$. Next, each component of the quintuple is explained.

Group Evolution: The *group evolution* of the annotations of e is represented by s in $q(e|i,i+1)$, and corresponds to the overall similarity of groups of annotations of e at time points i and $i+1$. The *group evolution* is computed as the similarity value assigned by *AnnSig* (see definition in Section 2.5) for the partition *PA* of the bipartite graph $BG = (A(e,i) \cup A(e,i+1), WE)$ and the ontology version O_{i+1}. The partition *PA* is computed by semEP, a graph

partitioning algorithm. Values close to 0.0 indicate that the annotations of e completely change during the two generations. Values close to 1.0 suggest that annotations either are maintained or are changed by similar terms.

For example, Figure 5.10 illustrates the bipartite graph between the annotations of protein `P48734` at 2010 and 2012. To compute the similarity values of *AnnSig*, semEP partitions the edges of this bipartite graph into six clusters or parts. Five out of these six clusters group the same GO term, e.g., GO:007067, and indicate that these five terms do not change in these generations. Further, the cluster that groups GO:0006468 and GO:00016310 indicates that the annotation GO:0006468 at 2010 changes into two terms GO:0006468 and GO:00016310 in 2012. Additionally, the GO term GO:0006916 is not similar to any other GO term in the 2012 generation of `P48734`, and similarly, GO terms GO:0006915 and GO:0043066 are dissimilar to the GO terms in the generation 2010. Therefore, there is no cluster that encloses these GO terms. During the computation of *AnnSig*, values of similarity of GO the terms in the same cluster are considered, as well as the number of terms that cannot be included in a cluster; thus, the *group evolution* of `P48734` at 2010 and 2012 is 0.95 indicating stability of the annotations of `P48734` during these generations.

Unfit Annotations: *Unfit annotations* are represented by n in $q(e|i,i+1) = (s,n,w,o,v)$, and are modeled as the normalized number of annotations of the generation $g(e,i)$ which are not included in any cluster of the partition PA produced to compute *AnnSig*. Formally, the number of *unfit annotations* is computed as $n = \frac{|a_n|}{|A(e,i)|}$, where:

$$a_n = \{t | t \in A(e,i) \wedge t \in O_i \wedge t \in O_{i+1} \wedge \forall p \in PA, \forall x \in A(e,i+1) \Rightarrow \nexists (t,x) \in p\}$$

Values close to 1.0 indicate that $A(e,i+1)$ does not contain most of terms in $A(e,i)$ or terms similar with them. Values close to 0.0 indicate that $A(e,i+1)$ contains either most of the terms in $A(e,i)$ or similar ones. For example, although the GO term GO:0006916 is not part of any cluster in the partition PA presented in Figure 5.10, GO:0006916 is neither included in the GO version of 2012. Thus, GO:0006916 does not meet condition in the above expression and is not an *unfit annotation*; the value of n is 0.0.

New Annotations: *New annotations* are represented by w in $q(e|i,i+1) = (s,n,w,o,v)$, and are modeled by the normalized number of terms contained in $A(e,i+1)$ that are in both ontology versions O_i and O_{i+1}, but are not part of $A(e,i)$ and are not included in any cluster of in the partition PA. The number of *new annotations* is $w = \frac{|a_w|}{|A(e,i+1)|}$, where:

$$a_w = \{t | t \in A(e,i+1) \wedge t \in O_i \wedge t \in O_{i+1} \wedge \forall p \in PA, \forall x \in A(e,i) \Rightarrow \nexists (x,t) \in p\}$$

Values close to 1.0 indicate that most of the terms in $A(e,i+1)$ are not in $A(e,i)$ and are dissimilar to them. Values close to 0.0 indicate that most of terms in $A(e,i+1)$ that are not in $A(e,i)$ are present in clusters formed by semEP and therefore their similarity is high. Although GO terms GO:0043066 and GO:0006915 are part of GO in 2010 and 2012, they are not annotations of `P48734` in the generation of 2010 and are not in any cluster of the partition PA reported in Figure 5.10. Thus, `GO:0043066` and `GO:0006915` meet the above expression, and both are *new annotations*; $w = \frac{2}{9} = 0.22$.

Obsolete annotations are represented by o in $q(e|i,i+1) = (s,n,w,o,v)$, and are measured as the normalized number of ontology terms used in $A(e,i)$ of the annotation dataset that are not available in the version O_{i+1} of the ontology. Formally, the number of *obsolete annotations* is computed as $o = \frac{|a_o|}{|A(e,i)|}$, where:

$$a_o = \{t | t \in A(e,i) \wedge t \in O_i \wedge t \notin O_{i+1}\}$$

Protein	Generations	s	n	w	o	v
P48734	2008-2010	0.95	0.0	0.14	0.0	0
	2010-2012	0.95	0.0	0.22	0.14	0.0
	2012-2014	**1**	**0.0**	**0.0**	**0.0**	**0.0**
	Aggregated	0.97	0.0	0.12	0.05	0.0
P06493	2008-2010	0.95	0.0	0.83	0.0	0.06
	2010-2012	**0.91**	**0.02**	**0.13**	**0.04**	**0.01**
	2012-2014	0.95	0.01	0.04	0.0	0.0
	Aggregated	0.94	0.01	0.33	0.01	0.02

Table 5.5: AnnEvol descriptions for proteins P48734 and P06493 among generations 2008-2010, 2010-2012, and 2012-2014.

Values close to 1.0 indicate that most of the terms in $A(e,i)$ are obsolete in O_{i+1}. For example, because GO:0006916 is not part of GO 2012, it is removed from the annotations of `P48734` in the generation 2012. Thus, GO:0006916 does meet the above expression and is considered an obsolete term; $o = \frac{1}{7} = 0.14$.

Novel Annotations: Finally, *Novel annotations* are presented by v in $q(e|i,i+1) = (s,n,w,o,v)$, and are measured as the normalized number of ontology terms used in $A(e,i+1)$ and that are not part of O_i; $v = \frac{|a_v|}{|A(e,i+1)|}$, where:

$$a_v = \{t | t \in A(e,i+1) \wedge t \notin O_i \wedge t \in O_{i+1}\}$$

Values close to 1.0 indicate that most of the terms in $A(e,i+1)$ are novel. Note that GO:0043066 and GO:0006915 cannot be considered novel because, though they do not belong to the annotations of `P48734` in the generation of 2010, they are part of GO 2010. Thus, there are no novel annotations, and $v = 0.0$.

Table 5.5 shows generation changes for proteins `P48734` and `P06493` between generations 2008-2010, 2010-2012, and 2012-2014, i.e., three quintuples are reported by both `P48734` and `P06493`. Rows describe the evolution of the annotations of an entity between two generations in terms of: *group evolution* (s), *unfit annotations* (n), *new annotations* (w), *obsolete annotations* (o), and *novel annotations* (v). Note that the *group evolution* is 1.0 for `P48734` in the generations 2012-2014, and the rest of the components are 0.0. This indicates no changes of the annotations in these generations. On the other hand, annotations of `P06493` in the generations 2010 and 2012 considerably change, i.e., two annotations in 2010 are removed in 2012 and two are included in 2012; one of the removed annotations is obsolete while one of the added annotations is novel.

The annotation sets evolve over the time. Thus, in a period $P = \{1,\ldots,n\}$ several generation changes denoted as $g(e|P) = [g(e,1), g(e,2), \ldots, g(e,n)]$. For example, to calculate aggregated values in Table 5.5 the generations of the proteins `P48734` and `P06493` in the time period P={2008, 2010, 2012, 2014} are considered. In this case, between two generations there is a distance of two-years, but a finer granularity may be considered. *Generation granularity* depends on the update frequency of the annotations of the dataset. Generations in $g(e|P)$ are ordered by date, so the oldest generation is the first in the list $g(e,1)$ and the most current is the last $g(e,n)$. Generation change sequences on a time period $P = \{1,\ldots,n\}$ are represented as a quintuple list Q containing one generation change for each pair of generations $Q(e|P) = [q(e|1,2), q(e|2,3), \ldots, q(e|n-1,n)]$, where $q(e|i,i+1)$ corresponds to the generation change of e between the generations $g(e,i) = (A(e,i), O_i)$ and $g(e,i+1) = (A(e,i+1), O_{i+1})$, e.g., generation changes of protein `P48734` on the time period P={2008, 2010, 2012, 2014} are represented as:

$$Q(\text{ P48734}|P) = \{q(\text{ P48734}|08,10), q(\text{ P48734}|10,12), q(\text{ P48734}|12,14)\}$$

Measuring Evolution of the Annotations of an Entity in a Time Period Given an ontology-based annotated entity e and a *sequence of generation changes* of e in a time period P=*{1,...,n}*, $Q(e|P)$, the evolution of e given P is represented as a quintuple $\bar{Q}(e|P)$ that summarizes the evolution of the annotations of e over the time period P:

$$\bar{Q}(e|P) =< \bar{S}, \bar{N}, \bar{W}, \bar{O}, \bar{V} >$$

- $\bar{S}$ represents the aggregated value of *group evolution* of the annotations of e in the period P, i.e., $\bar{S} = F(\{s|(s,n,w,o,v) \in Q(e|P)\})$. Values close to 1.0 indicate that exist a stable group of annotations that survived all the generation changes. Values close to 0.0 indicate that there does not exist such stable group and each generation change produce a total change in the annotation set. For example, consider the evolution of protein `P06493` in the time period *P={2008, 2010, 2012, 2014}* (Table 5.5), and suppose $F(.)$ is the arithmetic mean function, then aggregated value of the *group evolution* is 0.94 and indicates that a high number of annotations either are maintained during the time period P, or are changed by similar GO terms.

- $\bar{N}$ represents the aggregated value of *unfit annotations* of e in the period P, i.e., $\bar{N} = F(\{n|(s,n,w,o,v) \in Q(e|P)\})$. A value close to 1.0 means that most of annotations do not survive more than one generation, while a value close to 0.0 means that most of them survive or evolve into similar annotations. For example, the evolution of protein `P06493` in the time period *P={2008, 2010, 2012, 2014}* only few annotations are removed; thus, the arithmetic mean values of *unfit annotations* is low and corresponds to 0.01.

- $\bar{W}$ represents the aggregated value of *fit annotations* of e in the period P, i.e., $\bar{W} = F(\{w|(s,n,w,o,v) \in Q(e|P)\})$. A value close to 1.0 means that most of generations include a high proportion of new annotations that are not related with annotations in previous generation. A value close to 0.0 represents that most of generations contain a low proportion of new and different annotations. In the running example, new annotations are added to protein `P06493` in all the generations (Table 5.5); thus, the aggregated value of *fit annotations* is 0.12.

- $\bar{O}$ represents the aggregated value of *obsolete annotations* of e in the period P, i.e., $\bar{O} = F(\{o|(s,n,w,o,v) \in Q(e|P)\})$. Values close to 1.0 indicate an inattention in the annotation of the entity since most of annotations remain obsolete in most of generation changes. Values close to 0.0 mean that most of generations have most of annotations up to date in relation to the ontology.

- $\bar{V}$ represents the aggregated value of *novel annotations* of e in the period P, i.e., $\bar{V} = F(\{v|(s,n,w,o,v) \in Q(e|P)\})$. Values close to 1.0 indicate that most of annotations of most of generations are terms that were added in the last ontology version. Values close to 0.0 indicate that few novel terms are introduced in most of generation changes.

- $F(.)$ is the average function but can be substituted by any other aggregation function, for example, the median.

Based on the results presented in Table 5.5 and setting $F(.)$ as the arithmetic mean function, the aggregated evolution of `P06493` in the time period *P={2008, 2010, 2012, 2014}* is $\bar{Q}(P06493|P) =< 0.94, 0.01, 0.33, 0.01, 0.02 >$.

Interpretation of the Evolution of Annotations of an Entity

AnnEvol allows for the description of the following evolutionary properties of an ontology-based annotated entity e over a time period P:

- *Stable evolution of entity annotations:* Values of n and w of 0.0 suggest no significant changes in the annotation sets over a time period. Modified annotations are included in the clusters of *AnnSig* because they are

similar to annotations in the next generation, e.g., in Figure 5.10 GO terms GO:0006915, GO:0043066, and GO:0006915 are not part of any cluster because they are not similar enough to the other GO terms. Therefore, values of n and w are not 0.0 and suggest that the knowledge encoded in the annotations of protein P48734 is not completely stable during the generations 2010 and 2012. A value of s in the interval $(0.0, 1.0)$ may mean that: *i*) the annotation set $A(e_{i+1})$ is extended with terms similar to the already existing in $A(e_i)$; *ii*) the annotation set $A(e_i)$ is reduced but there are terms in $A(e_{i+1})$ which are similar to the removed; or *iii*) some terms in $A(e_i)$ are substituted by similar terms in $A(e_{i+1})$.

- *Retraction of significant knowledge*: A value of n higher than 0.0 means that some annotations are deleted, and there are no similar ontology terms in the current generation that represent this knowledge.

- *Addition of significant knowledge*: A value of w higher than 0.0 means that new annotations are introduced in the annotation set, and that there are no similar annotations in the previous generation. For example, knowledge encoded in GO terms GO:0043066, and GO:0006915 is added in generation 2012 of protein P48734.

- *Elimination of obsolete knowledge*: The component o of the quintuple allows us to identify how many ontology terms are obsolete in the more current version of the ontology. For example, because GO:0006916 is not part of GO 2012, the elimination of this annotation in the generation 2012 of protein P48734 corresponds to the elimination of *obsolete annotations*.

- *Improvements of knowledge*: Annotations can be improved in three ways: *i*) Evolution of the ontology and the definition of new ontology terms, to afford a better description of the entity in terms of their annotations. The component v indicates that novel ontology terms are used in the more current version of the entity annotations. *ii*) Annotations $A(e_i)$ can be extended in the version $A(e_{i+1})$ with new annotations related with those already present in $A(e_i)$. The relatedness between two ontology terms can be measure with a similarity measure like D_{tax}, GADES, etc. To discover this improvement, the components s, n and w of a quintuple $q(e|i,i+1)$ are considered. The component s measure the similarity between the partitions formed by *AnnSig* in $A(e_i)$ and $A(e_{i+1})$. A value of s close to 1.0 and values of n and w of 0.0 indicate that all the terms in $A(e_i)$ and $A(e_{i+1})$ are included in clusters of *AnnSig*, i.e., these annotations are similar to other annotations. *iii*) Annotations $A(e_i)$ can be extended in the version $A(e_{i+1})$ with new annotations not related to the annotations in $A(e_i)$. The component w indicates the number of annotations in $A(e_{i+1})$ that are not included in clusters of *AnnSig*, i.e., w indicates the terms in $A(e_{i+1})$ that are dissimilar to the terms in $A(e_i)$.

Annotation stability: Given a sequence of generation changes of an entity e on a time period P, $Q(e|P)$, the stability of its annotations is measured as the proportion of quintuples in $Q(e|P)$ that indicates no change in the annotation sets. A quintuple $q(e|i,i+1)$ reflects stability if its components follow the expression $q(e|i,i+1) = (1.0, 0.0, 0.0, o, v)$. The value of the component s indicates that the same partitions are found in the two annotation sets $A(e_i)$ and $A(e_{i+1})$, while the values of n and w suggest that no term is deleted and no new term is added. The combination of these three values guarantee that $A(e_i) = A(e_{i+1})$. This property ensures that both o and v are equal to 0.0. The stability of the annotations in a sequence of generation changes $Q(e|P)$ of an entity e on a time period P is modeled as $stab(Q(e|P)) = \frac{|U|}{|Q(e|P)|}$, where:

$$U = \{(s,n,w,o,v) \in Q(e|P) | s = 1.0 \wedge n = 0.0 \wedge w = 0.0\}$$

Dataset	2010	2012	2014
UniProt-GOA	12.21	14.69	16.13
SwissProt	5.39	8.11	8.66

Table 5.6: Average of annotations per protein

Knowledge monotonicity: Lower values of retraction of significant knowledge suggest higher evolution stability. The monotonicity of an annotated entity is defined as the proportion of quintuples in $Q(e|P)$ that only reflects addition of knowledge. A quintuple $q(e|i,i+1)$ exhibits *monotonicity* if it meets the following condition:

$$q = (s, 0.0, w, o, v),\ \textit{where}\ s > 0$$

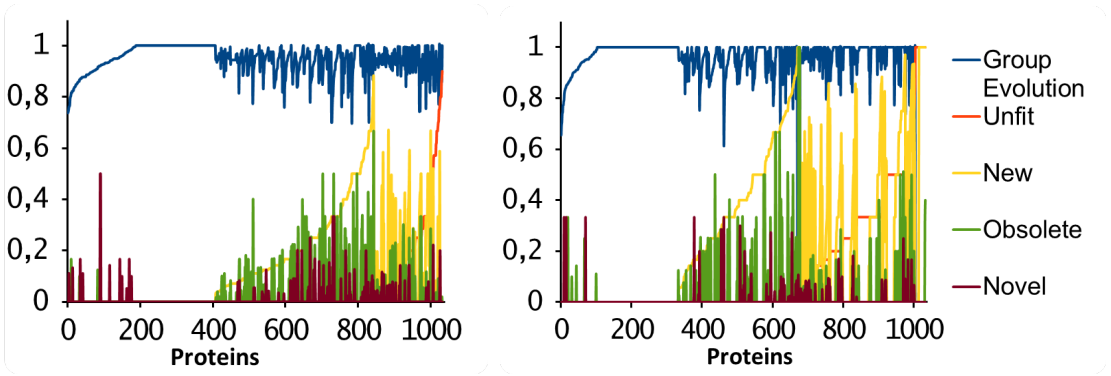
Thus, entities do not loose meaningful annotations. Some annotations may be lost, but they are similar to other annotations present in the most current generation, and therefore, this lost is not considered a retraction of significant knowledge.

Measuring Evolution of the Annotations of Entities in a Dataset Given a set of ontology-based annotated entities $E = \{e_1, e_2, ..., e_m\}$, the generation changes at time i of the entities in E are defined as $Dg(E|i) = \{g(e,i)|e \in E\}$, e.g., a set generation of UniProt-GOA in 2010 contains one generation $g(p|2010)$ for each protein p in the dataset. Given two set generations $Dg(E|j)$ and $Dg(E|k)$ and a period P from j to k, e.g., UniProt-GOA 2010 and 2012, the dataset generation changes is defined as $DQ(E|P) = \{q(e|j,k)|g(e,j) \in Dg(E|j) \wedge g(e,k) \in Dg(E|k) \wedge j \in P \wedge k \in P\}$ which contains generation changes $q(e|j,k)$ for each entity e in E in the time period P. Considering a dataset of two proteins $E = \{\texttt{P48734}, \texttt{P06493}\}$, two set generations, e.g., $Dg(E|2010)$ and $Dg(E|2012)$, and a time period P from 2010 to 2012, $DQ(E|P)$ contains all quintuples in each sequence of generation changes $Q(\texttt{P48734}|P) = [q(\texttt{P06493}|10,12)]$ and $Q(\texttt{P06493}|P) = [q(\texttt{P06493}|10,12)]$, the dataset generation changes corresponds to $DQ(E|P) = \{Q_{\texttt{P48734}} \cup Q_{\texttt{P06493}}\} = \{q(\texttt{P48734}|10,12), q(\texttt{P06493}|10,12)\}$. Similar to when *AnnEvol* is applied to evaluate the evolution at the level of entities, *AnnEvol* for a dataset E allows for aggregating the evolutionary properties of the entities in E observed over generations.

Experimental Study

AnnEvol is utilized to measure the evolution of the annotations of proteins in CESSM 2008 and CESSM 2014. For each set of proteins, three generations or annotation sets are considered: 2010, 2012, and 2014. There are two quintuples per dataset, one per generation change: 2010-2012, and 2012-2014. Table 5.6 reports on the number of annotations per protein for all the generations of each dataset. Observe that the average number of annotations for UniProt-GOA almost doubles the average of SwissProt. The explanation is that SwissProt contains only manually curated annotations, while UniProt-GOA additionally includes electronically predicted annotations.

Tendency of Changes in two Generations The goal of this experiments is to study the tendency of the changes of the annotations in the datasets SwissProt and UniProt-GOA in the generation change 2010-2012. Figures 5.11(a) and 5.11(b) reflect the values of each generation change of each protein in the datasets. Different behaviors can be distinguished in these datasets. X-axis contains one quintuple per proteins. Quintuples are sorted on ascending values of the five components in order n, w, s, o, v. Y-axis represents the values of each component of the quintuple, whose values are in $[0.0, 1.0]$. Tendency of *group evolution* is reported with a blue line. Observe that the blue line for SwissProt values has a larger amplitude than for UniProt-GOA and that this line stays longer with value 1.0. This indicates that fewer proteins change in SwissProt 2012. However, those proteins that change,



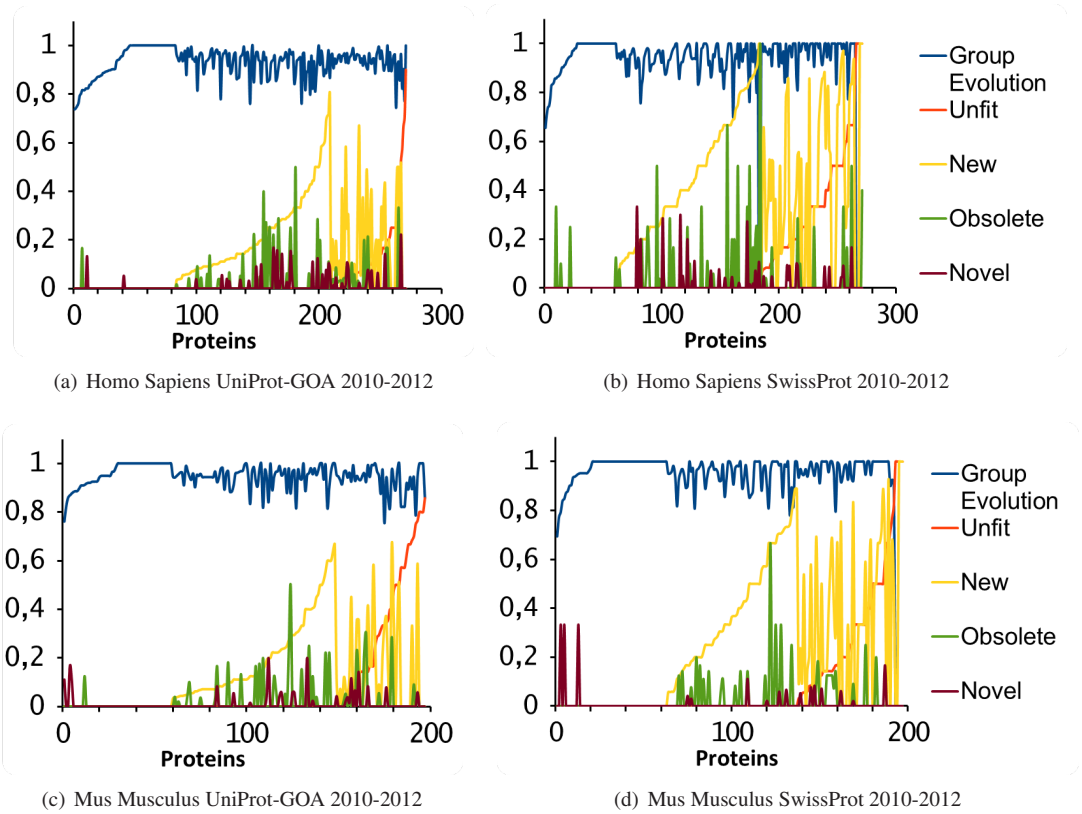
(a) UniProt-GOA 2010-2012 (b) SwissProt 2010-2012

Figure 5.11: Generation changes of 2010-2012

undergoes major changes than in UniProt-GOA. Therefore, changes in SwissProt are less equally distributed than in UniProt-GOA. Tables 5.7 and 5.8 support this statement. Table 5.7 shows that the average value of *group evolution* in SwissProt is lower than in UniProt-GOA. Results on Table 5.8 suggest that SwissProt is more stable than for UniProt-GOA in 2010-2012. Tendency of *unfit annotations* is described with an orange line. As can be observed in Figures 5.11(a) and 5.11(b), there are more proteins in SwissProt that experience removing of annotations (356 proteins in SwissProt versus 189 in UniProt-GOA). Moreover, the area under the orange line in SwissProt is larger than in UniProt-GOA, i.e., the number of removed annotations is proportionally greater in SwissProt. This is confirmed by Table 5.7 where the normalized number of *unfit annotations* in SwissProt is almost the triple than in UniProt-GOA. Tendency of *new annotations* is reported with a yellow line. Figures 5.11(a) and 5.11(b) show that the number of proteins were annotations were added is slightly greater in SwissProt (566 vs. 549 proteins in UniProt-GOA). Table 5.7 indicates that the normalized number of *new annotations* for the generation change 2010-2012 is about the double in SwissProt, with a value of 0.249, while UniProt-GOA has a *new annotations* value of 0.129. The amplitude of the yellow line is also larger in SwissProt. This suggests that new annotations are more uniformly distributed in UniProt-GOA than in SwissProt.

Tendency of *obsolete annotations* is described with a green line. There are more proteins in UniProt-GOA than in SwissProt 2010 that contain annotations that become obsolete in 2012 (171 versus 139 proteins). However, as the amplitude indicates, obsolete annotations are more equally distributed in UniProt-GOA and the proportional number of obsolete annotations per protein is lower in UniProt-GOA (0.026) than in SwissProt (0.033) (Table 5.7). Tendency of *novel annotations* is reported with a claret line. UniProt-GOA contains more proteins that include novel terms in their annotations (102 versus 69 proteins in SwissProt). The average of novel annotations per protein is also slightly higher in UniProt-GOA with a value of 0.009 versus 0.007 in SwissProt (Table 5.7). This suggests a tendency of using more novel terms in UniProt-GOA than in SwissProt. The combination of the results observed for *obsolete annotations* and *novel annotations* indicates that for the generation change 2010-2012 UniProt-GOA reacts quicker to the inclusion of new GO terms than SwissProt; however, UniProt-GOA seems to slower react to the deletion of terms in the ontology. Finally, Figures 5.12(a)-(d) illustrate the evolutionary behavior of the annotations of the organisms Homo Sapiens and Mus Musculus in UniProt-GOA and SwissProt for the generation change 2010-2012. As observed, proteins of these two organisms follow a similar behavior to the rest of the proteins of the studied datasets.

Evolutionary Behavior Next, aggregated values of *AnnEvol* are reported for each dataset. The behavior observed in the generation change 2010-2012 for *group evolution* can be generalized. Table 5.7 contains group evolution values of 0.956 and 0.949 for UniProt-GOA and SwissProt, respectively. Stability values presented



(a) Homo Sapiens UniProt-GOA 2010-2012

(b) Homo Sapiens SwissProt 2010-2012

(c) Mus Musculus UniProt-GOA 2010-2012

(d) Mus Musculus SwissProt 2010-2012

Figure 5.12: Generation Change 2010-2012 Homo Sapiens and Mus Musculus

in Table 5.8 show an even higher stability of the SwissProt with respect to UniProt-GOA than the observed in generation change 2010-2012. Therefore, though fewer proteins change their annotations in SwissProt than in UniProt-GOA, changes in SwissProt are stronger than in UniProt-GOA. Monotonicity aggregated values in Table 5.8 also confirms that even though the aggregated monotonicity value of SwissProt is higher than the observed in generation change 2010-2012, the monotonicity in UniProt-GOA is demonstrably higher than in SwissProt. Aggregated values contained in Table 5.7 confirm tendencies described for the generation change 2010-2012 with no remarkable changes.

Dataset	Gen.	Group Evol.	Unfit Annot.	New Annot.	Obsolete Annot.	Novel Annot.
UniProt	**10-12**	0.946	**0.044**	**0.129**	0.026	0.009
	12-14	0.966	0.027	0.064	0.006	0.006
	Aggr.	0.956	0.036	0.097	0.016	0.008
SwissProt	**10-12**	0.930	**0.127**	**0.248**	0.033	0.007
	12-14	0.968	0.050	0.077	0.014	0.003
	Aggr.	0.949	0.089	0.163	0.024	0.005

Table 5.7: Aggregated behavior over generation changes 2010-2012 and 2012-2014 for UniProt-GOA and SwissProt

Dataset	Generations	Stability	Monotonicity
UniProt-GOA	2010-2012	0.213	0.817
	2012-2014	0.418	0.816
	Aggregated	0.316	0.817
SwissProt	2010-2012	0.224	0.655
	2012-2014	0.518	0.790
	Aggregated	0.371	0.723

Table 5.8: Stability and monotonicity values for each generation transition

5.4 Integration of Graph Structured Data: Fuhsen

Public knowledge bases like DBpedia, Yago, or Wikidata describe real world entities in form of knowledge graphs in a Resource Description Framework (RDF). Using triples in the form of subject-predicate-object (s, p, o), RDF allow for describing different characteristics of a graph entity like the categories an entity belongs to; how is the entity related with the rest of entities in the graph; or which literals or attributes are related to the entity. Semantic Web and Linked Data initiatives have fostered the publication of data in this form and today there are more than 1,100 datasets [17] described with Linked Data vocabularies. This growth causes the same entity to be described with different vocabularies in different datasets. Semantic Web technologies like OWL offer solutions, such as the *owl:sameAs* predicate, to integrate the different descriptions of the same entity. Nevertheless, the curation of this kind of relations among the descriptions is a tedious and prone to errors task. Hence, an automatic approach to discover *owl:sameAs* relations is desired. In this section, a semantic approach to discover this kind of relations is presented. The approach, called FuhSen, makes use of a similarity measure to determine the similarity between two knowledge graph entities is presented. The evaluation shows the behaviour of the approach when using semantic and non-semantic similarity measures. In this section is shown, how feeding FuhSen with semantic similarity values computed with GADES improves the effectivenes of the integration approach. This work was published at the International Conference on Semantic Computing [13], at the International Conference on Web Intelligence, Mining and Semantics [14] and at the International Conference on Database and Expert Systems Applications [27].

5.4.1 Motivating Example

Linked Data initiatives foster the interconnection of Linked Data datasets to reduce redundancies, inconsistencies, and facilitate the access to the information from a single access point, i.e., without querying different knowledge bases. Due to the large amount of Linked Data datasets, is difficult for humans to maintain and keep updated the multiple knowledge bases. Therefore, a method able to automatically detect and integrate descriptions about the same real world entity in different knowledge bases is needed.

The increasing amount of knowledge bases causes the same real world entity to be described with different Linked Data vocabularies in different ways. Due to the lack of unique identifiers and the expected differences in the describing schemes, identifying when two descriptions refer to the same real world entity is a challenging task. Figure 5.13 shows a portion the descriptions of the painter Eugenio Bonivento in DBpedia, Wikidata and Oxford Art knowledge bases. Wikidata and DBpedia are knowledge bases of general domain and therefore describe the painter with common properties like the birth place or the name. However, Oxford Art is a domain specific knowledge base and includes much more information about the painter like his paintings and their properties. Further, literals or attributes can be described in different formats depending on the knowledge base. For example, the format the date of birth is different in DBpedia and Wikidata. Finally, each knowledge base follows a different

[17] `http://lod-cloud.net/`

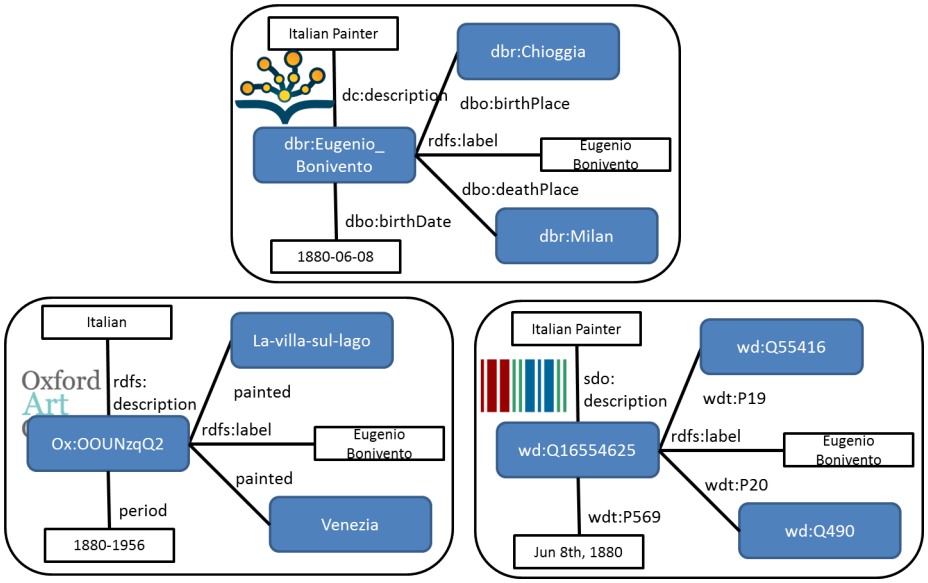


Figure 5.13: Descriptions of Eugenio Bonivento in DBpedia, Wikidata and Oxford Arts [13].

dbr:Chioggia	owl:sameAs	wd:Q55416
dbr:Milan	owl:sameAs	wd:Q490
dbo:birthPlace	owl:equivalentProperty	wdt:P19
dbo:deathPlace	owl:equivalentProperty	wdt:P20
dbo:birthDate	owl:equivalentProperty	wdt:P569

Figure 5.14: Logical axioms between DBpedia and DrugBank.

resource naming strategy. While DBpedia uses readable URIs, Wikidata and Oxford Art name resources with auto-generated identifiers hard to comprehend. These facts indicate that just a string similarity measure is not enough to estimate accurately the similarity between entities of different knowledge graphs.

Despite these differences, usually there is an overlapping among the descriptions that may allow for determining when they correspond to the same real world entity. This overlapping may not be easy to detect due to the different used vocabularies to describe the entities and the different levels of detail of the description. Logical axioms encoded in the graph may alleviate this problem bridging the gap among the used vocabularies. Figure 5.14 contains example of these axioms. Considering these axioms, it is possible to know that *wdt:20* and *dbo:deathPlace* represent the same property, or that *dbr:Milan* and *wd:Q490* represent the Italian city of Milan.

Syntactic approaches that rely on literals, like SILK [39] or LIMES [67], will fail detecting that the two descriptions in Figure 5.13 refer to the same real world entity. They may succeed if they only consider the `rdfs:label` predicate. Nevertheless, the restriction to the `rdfs:label` predicate must be manually specified and one cannot suppose that this strategy will succeed for all entities. Even taking into account the logical axioms, these approaches only consider one resource characteristic, the attributes, and they will not detect, e.g, that both descriptions in Figure 5.13 have a similar neighborhood and are in a close position in the hierarchy. Thus, a semantic approach able to consider multiple resource characteristics (the hierarchy, the neighborhood, etc.) is needed.

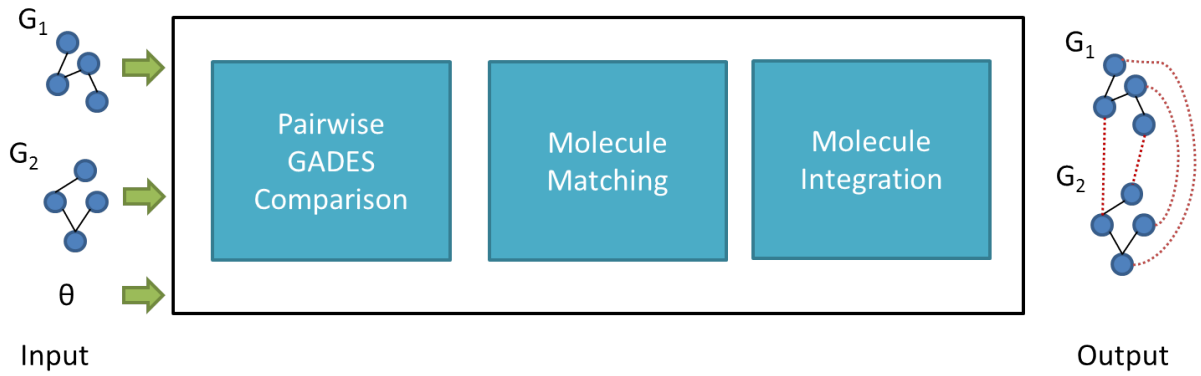


Figure 5.15: RDF Molecule Integration Approach Architecture [13].

5.4.2 FuhSen: A Semantic Integration Approach

Problem Definition

Let $G_1 = (V_1, E_1, L_1)$ and $G_2 = (V_2, E_2, L_2)$ be two knowledge graphs. Let F be the set of real world entities such that, for each RDF molecule in $\phi(G_1) \cup \phi(G_2)$, there is one real world entity in F. The problem of integrating two knowledge bases consists of identifying when two RDF molecules $M_1 \in \phi(G_1)$ and $M_2 \in \phi(G_2)$ describe the same real world entity $f \in F$, i.e., when an *owl:sameAs* relation exists between the molecules M_1 and M_2.

RDF Molecule Integration Approach

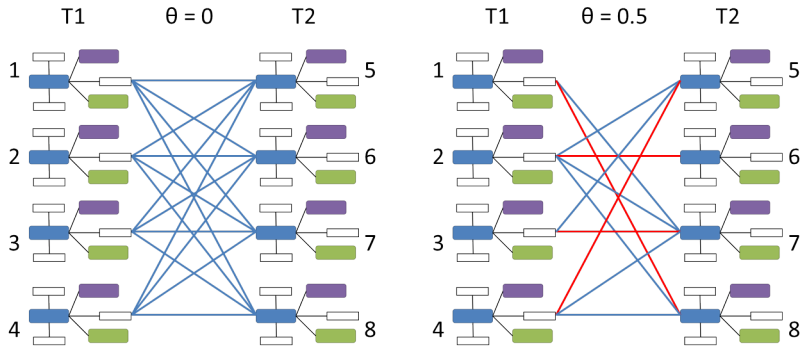
Figure 5.15 shows the architecture of the RDF molecule integration approach. The approach receives four inputs: Two knowledge graphs containing RDF molecules described with different Linked Data vocabularies, a similarity measure for RDF molecules Sim and a threshold $\theta \in [0, 1]$. The output is the set of discovered *owl:sameAs* relations among the given RDF molecules.

The approach consists of three phases. First, the RDF molecules are extracted from the knowledge graph. Let T_1 and T_2 be the molecules extracted from $\phi(G_1)$ and $\phi(G_2)$ respectively. Then, a pairwise comparison is performed with Sim among the extracted molecules in T_1 and T_2.Once the Sim values are available, a bipartite graph like the showed in Figure 5.16 is built. Edges in this bipartite graph are weighted with the computed similarity values. In order to reduce the amount of produced false positives, edges with a weight lower than the provided threshold θ are removed from the bipartite graph. Then, the maximum 1-1 weighted matching, represented with red edges in Figure 5.16, is computed. Edges contained in the 1-1 perfect matching connect knowledge graph entities that are considered *semantically equivalent*, i.e., that represent the same real world entity. These edges correspond to the discovered *owl:sameAs* relations.

5.4.3 Empirical Evaluation

An empirical evaluation is conducted to address the research question of which is the impact of considering semantics when integrating graph structured data. Thus, the FuhSen approach is given as input semantic (GADES and GBSS [72]) and non-semantic similarity measures (Jaccard). GADES and Jaccard can be used in any kind of knowledge graph, while GBSS is tailored for DBpedia entities.

Benchmark: The benchmark consists of two experiments involving DBpedia and Wikidata knowledge bases. In the first experiment 500 molecules are extracted from the live version of DBpedia (July 2016). This experiment shows that FuhSen is able to detect when two descriptions in the same knowledge graph corresponds to the same



	1	2	3	4
5	0.1	0.38	0.73	0.8
6	0.2	0.9	0.47	0.2
7	0.5	0.62	0.86	0.64
8	0.8	0.55	0.18	0.7

	1	2	3	4
5	0	0	0.73	0.8
6	0	0.9	0	0
7	0.5	0.62	0.86	0.64
8	0.8	0.55	0	0.7

Figure 5.16: Bipartite graphs of RDF molecules [13].

real world entity. In the second experiment, 20,000 real world entities are selected and their descriptions extracted from DBpedia and Wikidata. Hence, the effectiveness of FuhSen when integrating molecules from general domain knowledge bases is shown.

Metrics: Precision, recall and F-measure are used to measure the effectiveness of FuhSen with the different similarity measures in the three experiments. Let E_{GS} be the set of *owl:sameAs* edges in the gold standard. Let E_{FuhSen} be the set of *owl:sameAs* edges discovered by FuhSen. Precision is measured as the fraction of correctly discovered edges:

$$\text{Precision} = \frac{|E_{GS} \cap E_{FuhSen}|}{|E_{FuhSen}|}$$

Recall measures the fraction of discovered correct edges:

$$\text{Recall} = \frac{|E_{GS} \cap E_{FuhSen}|}{|E_{GS}|}$$

The F-measure metric is computed as the harmonic mean of the precision and the recall:

$$\text{F-measure} = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Integration of DBpedia molecules

The goal of this experiment is to evaluate the impact of considering semantics when integrating molecules described with the same vocabulary but with different properties.

Benchmark: 500 molecules of type Person are extracted from the live version of DBpedia (2016). Based on the original molecules, two datasets D1 and D2 are created by randomly deleting and editing triples of the molecule. Hence, for each original molecule two different descriptions are generated. Figure 5.17 shows an example of molecule generation. Editions may affect triple predicates or objects. Both can be replaced by some of their ancestors, i.e., predicates or entities that are above them in the respective hierarchies. Thus, no noise is introduced in the molecules, but the information or specificity is reduced. D1 describes the 500 molecules with 17,951 triples, while D2 contains 17,894.

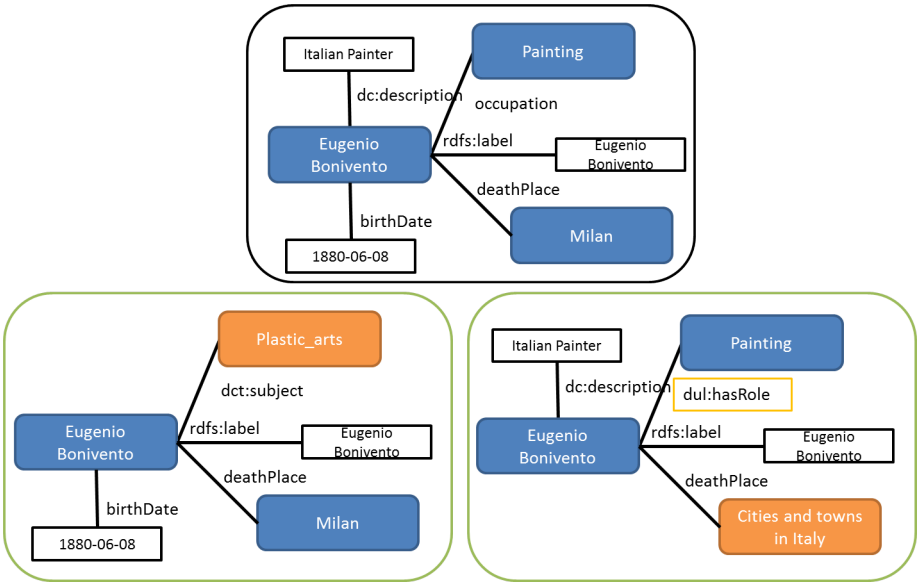


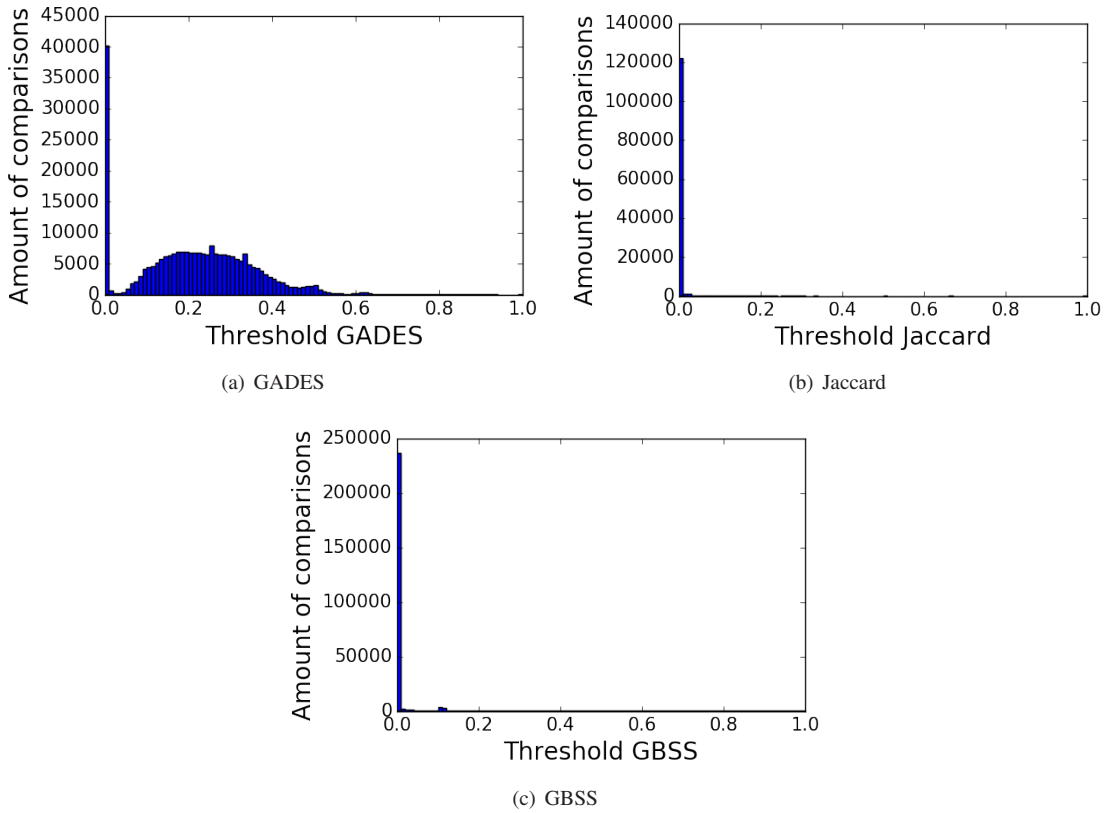
Figure 5.17: Random deletion and edition of triples in RDF molecules

	GADES			Jaccard			GBSS		
θ	Prec.	Rec.	F-Me.	Prec.	Rec.	F-Me.	Prec.	Rec.	F-Me.
NT	0.81	0.81	0.81	0.77	0.78	0.78	0.47	0.47	0.47
P95	0.84	0.81	0.82	0.78	0.78	0.78	0.91	0.46	0.61
P97	0.84	0.81	0.82	0.91	0.78	0.84	0.92	0.46	0.61
P99	0.86	0.76	0.8	0.91	0.77	0.83	0.94	0.38	0.54

Table 5.9: FuhSen Effectiveness on DBpedia.

Baseline: The set of real world entities F corresponds in this experiment to the original 500 molecules extracted from DBpedia. Let $f \in F$ be one of the original molecules. Let m_1 and m_2 be the two molecules generated from f. The gold standard consists of the set of *owl:sameAs* relations that connect the each pair of molecules m_1, m_2 such that both molecules are generated from the same molecule $f \in F$.

GADES, Jaccard, and GBSS are used within the FuhSen approach to discover *owl:sameAs* relations. GADES is implemented as the average among Sim_{hier}, $\text{Sim}_{\text{neigh}}$ and Sim_{attr}. The Jaccard similarity among two molecules $m_1 = (s_1, T_1)$, $m_2 = (s_2, T_2)$ is defined as the Jaccard coefficient of their tails, i.e., $\frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$. Table 5.9 shows the precision, recall and F-measure of the three measures. For each measure, three thresholds corresponding to the 95th, 97th, and 99th percentile are applied. Instead of applying a threshold with an absolute value, percentiles are used in this experiment. Further, a row indicating the effectiveness when no threshold (NT) is applied is included. The value distribution of each similarity measure is very different with respect to the others. Figure 5.18 contains the histograms of the similarity values returned by each measure. GBSS and Jaccard returns 0 for most of the comparisons. Nevertheless, GADES returns higher values for some comparisons. Thus, setting the threshold with an absolute value would lead to an unfair comparison discarding a larger amount of matchings for GBSS and Jaccard than for GADES. Setting the threshold with the same percentile for each similarity measure ensures to discard and retain a similar amount of RDF molecule matchings. The best F-measure value is reached by Jaccard applying the 97th percentile as threshold. GADES obtains the best results when no threshold is applied. Jaccard is able to outperform GADES because molecules are described with the same vocabulary, i.e., molecules are homogeneously described and there is no need to explore the knowledge graph to find similarities among properties or knowledge graph entities.



(a) GADES

(b) Jaccard

(c) GBSS

Figure 5.18: Histogram of the similarity scores between GADES, Jaccard, and GBSS for DBpedia molecules with different threshold values

5.4.4 Integration of DBpedia and Wikidata Molecules

The goal of this experiment is to evaluate the impact of considering semantics when integrating molecules described with the with different vocabularies, but the same level of specificity or abstraction.

Benchmark: 20,000 molecules of type Person[18] are extracted from the live version of DBpedia (2016) and Wikidata. In this case the molecules are already different, so it is no necessary to delete or edit their triples. The 20,000 molecules are described with 1,421,604 and 855,037 in DBpedia and Wikidata respectively.

Baseline: DBpedia and Wikidata are two well connected datasets, i.e., a large amount of entities are already connected through *owl:sameAs* relations. Particularly, the 20,000 DBpedia molecules involved in this experiment are connected with this predicate to their equivalent description in Wikidata. These edges compound the gold standard in this experiment.

Table 5.10 contains the results of GADES and Jaccard in this experiment. Values of θ correspond to the percentiles 95, 97, and 99 of the corresponding similarity measures. Further, the No Threshold (NT) row indicates the results of both measures when no threshold is applied. In this case, GADES obtains better results than Jaccard regardless the applied threshold. The maximal F-Measure value obtained by GADES is 0.76, while Jaccard only gets 0.266. Thus, GADES is more suitable than Jaccard when molecules are described with different vocabularies.

[18] https://github.com/RDF-Molecules/Test-DataSets/tree/master/DBpedia-WikiData/20161006

θ	GADES			Jaccard		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
NT	0.76	0.76	0.76	0.253	0.253	0.253
P95	0.836	0.588	0.69	0.253	0.253	0.253
P97	0.861	0.537	0.661	0.253	0.253	0.253
P99	0.913	0.39	0.547	0.282	0.252	0.266

Table 5.10: FuhSen Effectiveness on DBpedia gand Wikidata Molecules.

5.5 Conclusions

In this chapter the benefits of considering semantics for enhancing knowledge graph quality are shown. Particularly, three knowledge graph quality issues are addressed: relation discovery, evolution analysis and integration. Thus, a framework including approaches able to cover these knowledge graph issues is presented. Hence, given two knowledge graph entities, the framework helps to decide a) when these two entities should be related according to a certain property in the graph (relation discovery), b) if one of these entities is the evolution of the other one according to their neighborhoods (evolution analysis), and c) if these two entities are actually different descriptions of the same real world entity and should be integrated (integration).

The relation discovery task is solved by KOI, an approach that exploits semantics and graph structure information in order to discover missing relations among two entities in a knowledge graph. KOI considers semantics encoded knowledge graphs by taking into account two resource characteristics: the attributes and and the neighborhood. Thus, KOI is able to identify portions of the knowledge graphs containing highly similar entities with highly similar neighborhoods where to discover new relations. Reported experimental results suggest that KOI outperforms state-of-the-art approaches that: i) do not consider semantics (KNN TFIDF) or ii) do not identify graph portions containing highly similar entities (KNN D2VN and METIS). Though knowledge graphs contain different types of entities and relations, KOI is only able to discover relations among entities of the same type. Further, experiments show that KOI is able to outperform state-of-the-art approaches in knowledge graphs where entities are described with textual information. In the future, other domains including knowledge graphs with entities described in a different way, e.g., images or gene sequences, may be studied.

The evolution analysis issue is addressed by AnnEvol, a framework to analyze the evolution of annotation graphs. AnnEvol enables user to identify when significantly annotations are added or removed to the annotation graph, when annotations become obsolete because of changes in the ontology, or when recently added ontology terms are included in the annotation graph. Based on these metrics, AnnEvol also delivers complex knowledge and informs the user about the stability or the monotonicity of the annotations of a certain entity. Thus, curators can recognize anomalies in the evolution of entities and keep annotation graphs updated. AnnEvol exhibits a good performance with proteins annotated with GO terms. GO is a specific ontology from the biological domain with a well defined structure, i.e., with a deep hierarchy and an object property hierarchy. Similarly to KOI, AnnEvol may be evaluated in the future in knowledge graphs with different characteristics, in which the context also matters. For example, AnnEvol may be used to determine the evolution of customer opinions regarding a certain product over the time. Further, the results obtained with AnnEvol in form of metrics may be used as input for other approaches, like machine learning approaches, in order to discover patterns in the evolution of such knowledge graph entities.

Finally, the impact of considering semantics when integrating descriptions of entities from different knowledge graphs is shown through FuhSen. FuhSen integrates RDF molecules from heterogeneous knowledge graphs with the help of a similarity measure. Experimental results show that GADES, a semantic similarity measure that takes into account several resource characteristics, enables FuhSen to get better results than other semantics (GBSS [72]) and non semantic similarity measures (Jaccard). FuhSen exhibits a good performance when integrating descrip-

tions of knowledge graph defined with the same or similar vocabularies and which cover the same aspects of the entities. In the future, FuhSen may be evaluated in different scenarios, e.g., integrating knowledge graph entities that describe the same real world entity from a general and a domain specific perspective.

Knowledge graphs facilitate the representation of data in a flexible way without need of specifying a fixed schema. Further, knowledge graphs encode semantic knowledge in form of semantic annotations. Semantics allows computers to better understand the data they are working with by reducing the ambiguity and adding meaning to the relations among the described entities. Nevertheless, knowledge graphs also suffer from quality issues that may prevent applications that rely on them to succeed. Quality issues can be classified into two categories: correctness and completeness. In this chapter three knowledge graph quality issues are addressed through a framework that makes use of semantic similarity measures. While AnnEvol does not make any assumption about the correctness of the knowledge graph, KOI and FuhSen make use of the information included in the graphs in order to alleviate the incompleteness issue. Hence, the effectiveness of both approaches may be affected by the presence of incorrect information. In the future these framework can be extended to also cover incorrectness issues and study the impact of considering semantics when detecting incorrect information encoded in knowledge graphs.

6 Conclusions and Future Work

Because of to the Semantic Web and Linked data initiatives, the amount of data semantically described available on the Web has been increasing over the last years. Semantic technologies provide mechanisms to allow data be self-describing, i.e, the meaning of a data is implicit and it is no necessary to query third sources to understand the meaning of the data. The main goal of this thesis is to study the benefits of considering semantics in similarity measures and how these semantic similarity measures can alleviate knowledge graph quality issues. Hence, a similarity measure framework including similarity measures able to take advantage of such semantic information is defined. Further, the benefits of considering semantics through the measures in the developed framework are studied on three knowledge graph quality issues. This thesis provides the following main contributions:

1. Identification of the relevant resource characteristics to determine similarity values.
2. Definition of a similarity measure framework containing similarity measures able to consider such resource characteristics in an progressive way.
3. Computational complexity study of the defined measures to prove the suitability for real-time applications.
4. Application of the defined semantic similarity measure framework on three data-driven tasks to enhance knowledge graph quality. These tasks are relation discovery, evolution and integration.

A discussion of the presented contributions is introduced in Section 6.1. Section 6.2 presents the overall conclusions of this work and Section 6.3 gives an overview about open issues.

6.1 Discussion of Contributions

Research Question 1. *What is the improvement in terms of accuracy of considering semantics during the computation of similarity between two knowledge graph resources?*

In this thesis the relevant resource characteristics to determine the similarity between two knowledge graphs entities are identified. These characteristics are a) the relative hierarchical position of the compared entities, b) their neighborhoods, c) the shared information in terms of co-occurrence in a certain corpus and d) the attributes or literals associated to these resources. Previous works consider these characteristics in an isolated way and do not combine them to determine similarity values among knowledge graph entities. In this work, a semantic similarity measure framework is defined. The framework consists of four similarity measures able to consider and to combine these resource characteristics. The four similarity measures are evaluated on different datasets, being both versions of CESSM [79], a common benchmark for all of them.

The first and simplest similarity measure of the framework is OnSim. OnSim considers both the hierarchy and the neighborhood of knowledge graph entities as relevant resource characteristics. OnSim delivers more accurate values than similarity measures that only considers the hierarchy in an isolated way. Thus, OnSim is only outperformed by GI and RB, two similarity measure that only considers the shared information characteristic.

The second similarity measure in the framework is IC-OnSim, an extension of OnSim. IC-OnSim considers the hierarchy, the neighborhood and the shared information as relevant resource characteristics when computing similarity values. Hierarchies are organized in branches and not all the branches are described with the same level

of detail. Thus, similarity measures based on hierarchies may fail determining similarity values. The shared information reflects how similar are two entities based on their use in a certain corpus and can be measured in different ways. IC-OnSim determines the shared information among two terms based on the Information Content [82] of their Lowest Common Ancestor. The evaluation in CESSM shows that IC-OnSim outperforms OnSim and the rest of measures included in the benchmark with respect to four of the six gold standards.

In third place, GADES, a semantic similarity measure that considers the four resource characteristics and meets the metric properties is described. While OnSim and IC-Onsim are designed for ontology terms and OWL semantics, GADES is defined for knowledge graph entities and does not consider inferred knowledge. Hence, GADES does not need the execution of a semantic reasoner and, therefore, its computational complexity is lower. Unlike IC-OnSim, GADES determines the shared information among two entities based on their co-occurrences in the neighborhoods of other knowledge graph entities. GADES outperforms IC-OnSim and the rest of similarity measures in CESSM in three of the six gold standards and delivers competitive results for the other three ones. Additionally, GADES is evaluated on two benchmarks of texts annotated with DBpedia entities (Lee50 and STS), being in both of them the most correlated similarity measure with respect to the corresponding gold standards.

The combination of the information provided by the different resource characteristics is in the case of OnSim, IC-OnSim and GADES hand made. Thus, after a try and error process, a certain aggregation or combination function is found by the user or domain expert. GARUM extends GADES and alleviates the domain expert task by combining the different resource characteristic measures automatically means a machine learning based method. Hence, GARUM just needs a training dataset from which to learn how the different measures have to be combined in order to get accurate similarity values. GARUM is trained for four of the six CESSM gold standards and outperform the rest of measures included in the benchmark. Further, GARUM is also trained on on one of the text comparison benchmarks and obtains the most accurate results with respect the human based gold standard.

Answer to Research Question 1. Ontologies and knowledge graphs encode semantics by means different resource characteristics. The results obtained by the different described similarity measures show that combining the information encoded in each resource characteristic allow to compute more accurate similarity values. However, the importance of each resource characteristic may vary from domain to domain, and so the way of combining the different similarity values. The accuracy of the similarity measures is estimated according to the Pearson's coefficient with respect to a certain gold standard. GADES is able to outperform state-of-the-art approaches in both CESSM and Lee50, obtaining correlation coefficients 2.9% and 1.3% greater than the respective most correlated approaches. GARUM learns better aggregation functions than the defined for GADES and obtains correlation coefficients 16.6% and 2.6% greater than the obtained by the best state-of-the-art approaches in CESSM and Lee50, respectively.

Research Question 2. *How can semantic similarity measures efficiently scale up to large datasets and be computed in real-time applications?*

Together with the increasing amount of stream data fostered by initiatives like the *Internet of Things*, several applications that require the processing of these data in real-time are becoming relevant to the industry. If a similarity measure aims to be widely used, it has to be competitive when dealing with real-time applications in the performance-accuracy trade-off. OnSim and IC-OnSim are designed to compare ontology terms and make use of semantic reasoners to infer new facts in the ontology. These facts allow for improving the accuracy of the similarity values. Nevertheless, the use of semantic reasoners is costly in terms of time, reaching worst case complexities of 2NEXP [45] for the OWL2 entailment regime. OWL2 profiles lighten this problem by taking into account portions of the whole OWL2 language, i.e., considering lighter entailment regimes, and thus reducing the complexity of reasoning tasks. According to the empirical results, OnSim and IC-OnSim are able to perform in the best case around 369.97 comparisons per seconds. Nevertheless, GADES does not need the execution of a semantic reasoner. Hence, the computational complexity of GADES is lower and allow it to compare, in the same

dataset as OnSim and IC-OnSim, around 1598.8 comparisons per second. This velocity should be enough for near real-time applications as web applications.

Answer to Research Question 2. Trading expressiveness for velocity is needed to make semantic similarity measures suitable for real-time applications. Execution times obtained by OnSim and IC-OnSim when considering the full OWL2 entailment regime are not compatible with the processing of real-time or stream data. The reduction of the expressivity by applying one of the OWL2 profiles alleviates this problem and, depending on the ontology, the price to pay in terms of accuracy for such a speedup may be low. The results obtained with GADES even show that the complexity may be reduced by no considering reasoning at all, while the similarity measure remains competitive in terms of accuracy. GADES can be executed even in less time than OnSim and IC-OnSim with OWL2 profiles and can be used within real-time applications, e.g., recommendation systems in web applications. Further, empirical results exhibit a non-linear correlation among the expressiveness level considered by a similarity measure and its accuracy. Results of GADES in both versions of CESSM are slightly below the obtained by IC-OnSim. However, GADES exhibits a speedup coefficient of 4.32 and 5.04 for each CESSM version respectively. Hence, it seems that considering such a high level of expressiveness is not worthy to determine similarity values among entities in knowledge graphs.

Research Question 3. *What is the impact of using semantic similarity measures on data driven tasks, e.g., to discover relations in a knowledge graph?*

Data driven tasks as relation discovery or data integration require domain knowledge to be solve. Semantic Technologies enable data to be self-describing. This means, that a portion of the domain knowledge is already included in the data. Methods able to consider this semantic information may obtain better results in such data-driven tasks. In this work semantic-aware data-driven approaches for knowledge graph quality enhancement are described and evaluated. First, KOI a semantic-aware method to discover relations in a knowledge graph is defined. KOI considers the attributes and the neighborhood, two of the identified resource characteristics in the first research question. Further, FuhSen, an integration approach for Linked Data datasets defined by Collarana et al. [13], is fed with semantic similarity values computed with GADES. Thus, both approaches, KOI and FuhSen, depend on semantic descriptions to be successful. Hence, the effectiveness of both approaches may be affected when dealing with knowledge graphs suffering of incorrectness issues. In order to monitor the quality of these semantic descriptions, this thesis includes AnnEvol, a methodology to monitor annotation datasets that helps to find why are semantic-aware approached not working on a certain knowledge graphs.

Answer to Research Question 3. Empirical results of the semantic-aware data-driven approaches described in this thesis show that considering semantics during the execution of such data-driven tasks allows to outperform state-of-the-art approaches. KOI is able to discover relations obtaining an F-Measure 0.12 units above the best state-of-the-art approach. AnnEvol is able to describe the evolution of knowledge graph entities in a set-wise way according to their neighborhoods. FuhSen exhibits a good performance integrating knowledge graph entities either defined with the same vocabulary or describing the entities from the same perspective. On average, FuhSen obtains an F-Measure 0.27 points above the best approaches in the state-of-the-art. Nevertheless, these experiments were performed in knowledge graphs curated by humans with a well defined structure and a high level of correctness. Empirical results suggest that considering semantics in this kind of graphs increases the effectiveness of the approaches. However, two of these methods, KOI and FuhSen, rely on the information encoded in these knowledge graphs and, therefore, the presence of incorrect information may reduce the benefits of considering such semantics. Thus, the positive impact of semantics can be only ensured for knowledge graphs with low levels of incorrectness and well defined structures like DBpedia or GO.

6.2 Overall Conclusions

This work is motivated by the raising amount of semantically described data and the lack of similarity measures able to combine the different aspects offered by such semantic descriptions. Existing semantic similarity measures consider single resource characteristics to determine similarity values among knowledge graph resources and are not able to clearly outperform classic similarity measures like Jaccard. This work addresses three aspects about semantic similarity measures.

First, this work reveals the the need of considering and combining several resource characteristics to determine similarity values. The defined similarity measures show how they are able to deliver more accurate results when they combine simultaneously a larger amount of the identified resource characteristics. Nevertheless, the relevance of each resource characteristic depends on the domain. This thesis also shows that the relevance of each resource characteristic can be determined with the help of machine learning methods and a training dataset, which reduce the effort of domain experts during the implementation of the measure.

Second, this thesis shows how semantics and near real-time technologies can be matched as long as the expressivity level does not overcome a certain threshold. The high computational complexity of reasoning tasks in high expressive ontologies may prevent the combination of such tasks with real-time technologies. In Chapter 4 is shown how reducing the expressivity of considered semantics or even removing reasoning tasks, semantic similarity measures can be computed in polynomial time and used in near real-time applications while staying in competitive levels in terms of accuracy.

Finally, the positive impact of considering semantics is shown in Chapter 5. Relation discovery in knowledge graphs and integration of Linked Data datasets benefits from the semantics considered by the defined similarity measures. The effectiveness of theses tasks depends on the quality of the semantic descriptions. Therefore, this thesis also proposes a method to monitor the evolution of annotation datasets and to warn curators about possible quality issues.

6.3 Outlook

According to the evolution of the Linked Data Cloud, the amount of semantically described data will be larger and more varied in the future. Hence, the heterogeneity of the Linked Data Cloud will increase and knowledge graphs with different characteristics will appear. Particularly, we expect knowledge graphs differently created with respect to the current ones, which are created and curated either by large communities or experts. Thus, in the future we will see knowledge graphs maintained by smaller communities or even in an automatic way. These knowledge graphs may suffer more frequently of several quality issues as incorrectness. Finally, we also expect to have more knowledge graphs describing contextual aspects (e.g., time) requiring semantic approaches to deal with context awareness. Next, these issues are described with more detail.

Knowledge graph variety. Similarity measures and approaches in this thesis exhibit a good performance in structured and curated knowledge graphs. The considered ontologies GO and DBpedia are created and curated either by experts or large communities, which reduces the risk of containing incorrect information. However, it is not clear how these approaches will deal with knowledge graphs with other other characteristics as lighter structure (e.g., with a flatter class hierarchy or without property hierarchy), severe quality issues (incompleteness and incorrectness) or knowledge graphs describing more complex information as images or protein sequences.

Knowledge Graph Incorrectness. In this thesis semantic similarity measures are used to enhance knowledge graph quality. Nevertheless, completeness is the only quality aspect considered. As mentioned in other parts of this work, knowledge graph may also include incorrect information that do reflect properly the real world.

Incorrectness may have different causes as human action or technical issues and can be difficult to detect in large knowledge bases. In the future, the framework described in this thesis may be extended to also tackle incorrectness issues. Semantic descriptions of the data may assist the detection of inconsistencies or incorrectness issues and even help to repair them. Thus, maintenance effort of such knowledge bases would be reduce and, at the same time, the quality of their data would increase.

Similarity measures and context awareness. The semantic similarity measures defined in this thesis return values considering different characteristics of the compared entities. However, no contextual information (e.g., information about the agent requiring the computation of the similarity) is considered. Thus, the defined semantic similarity measures may be extended in order to, besides semantics, be able to take into account the context during the computation of similarity values. Hence, similarity values will vary depending on the time, the location or other contextual aspects. This would allow recommendation systems and search engines to also benefit from semantics. A clear example resides on the e-commerce domain. In order to maximize incomes, e-Commerce websites try to show and offer their customers the most interesting products for them. At the same time and for the same reason, they use their best efforts to facilitate the product search. Search and recommendation do not rely only on the product information, but also on the user profile and the context (time, location, recent searches, etc.). Only approaches able to deal with both, product information and customer profile, may succeed in such an scenario.

According to the evolution of the Linked Data Cloud, the amount of semantically described data is expected to be larger and more varied in the future. Hence, search engines and recommendation systems will need to handle larger amount of data without deteriorating performance or quality. Further, the use of machine learning algorithms in the industry is augmenting and, as described in this thesis, there are multiple approaches based on similarity measures. Thus, we expect the demand of similarity measures able to consider semantics to increase in the upcoming years.

A Appendix: Proof of Metric Properties for GADES

In this section the properties required for a metric are proved for GADES. These measures are defined as an aggregation function of up to four individual similarity measures: Sim_{hier}, $\text{Sim}_{\text{neigh}}$, $\text{Sim}_{\text{shared}}$ and Sim_{attr}. Each individual similarity measures computes a similarity value by considering one resource characteristics:

- Sim_{hier} computes the similarity among two knowledge graph entities according to their relative position in the hierarchy.
- Sim_{neigh} determines the similarity value between two knowledge graph entities based on how similar are their neighborhoods.
- $\text{Sim}_{\text{shared}}$ estimates similarity values based on the co-occurrences. Finally,
- Sim_{attr} considers the attributes of literals of the compared entities to determine how similar they are.

Supposing the aggregation function to be commutative and associative, the similarity measure will satisfy the required conditions for a metric whenever the individual aggregated elements also fulfill these conditions. According to the definition given in Section 2.1, a distance function d is a metric if fulfills the following properties:

- **Non-negativity**: The distance among two elements $a,b \in X$ cannot be negative $d(a,b) \geq 0.0$.
- **Identity**: The distance among two elements $a,b \in X$ is 0.0 iff they are the same element $d(a,b) = 0.0 \iff a = b$.
- **Symmetry**: The distance between two elements $a,b \in X$ is symmetric $d(a,b) = d(b,a)$
- **Triangle inequality**: The distance between two elements $a,c \in X$, $d(a,c)$ is minimal, i.e., for all element $b \in X$ the expression $d(a,c) \leq d(a,b) + d(b,c)$ applies.

Next, the metric properties are proved for the above four considered elements, i.e., Sim_{hier}, $\text{Sim}_{\text{neigh}}$, $\text{Sim}_{\text{shared}}$ and Sim_{attr}.

Sim_{hier} Let $G = (V,E,L)$ be a knowledge graph, $H \subseteq L$ the set of labels of hierarchical properties and $E_H = \{(v,r,w) \in E \mid r \in H\}$ the set of edges whose labels are in H. Sim_{hier} is defined as follows:

$$\text{Sim}_{\text{hier}}(x,y) = \begin{cases} 1 - D_{\text{tax}}(x,y) \\ 1 - D_{\text{ps}}(x,y) \end{cases}$$

where $x,y \in V$ are vertexes in the graph.

In order to prove that Sim_{hier} is a metric, it is necessary to prove that D_{tax} and D_{ps} satisfy the four properties of metrics. The definitions of D_{tax} and D_{ps} are as follows:

$$D_{\text{tax}}(x,y) = \frac{d(lca(x,y),x) + d(lca(x,y),y)}{d(root,x) + d(root,y)}$$

$$D_{\text{ps}}(x,y) = 1 - \frac{d(root,lca(x,y))}{d(root,lca(x,y)) + d(lca(x,y),x) + d(lca(x,y),y)},$$

where:

- $d(x,y)$ represents the number of hops of the shortest path between nodes x and y in the knowledge graph $G_H = (V, E_H, H)$,
- $lca(x,y)$ is a function $lca : V \times V \to V$ that represents the Lowest Common Ancestor of the nodes x and y,
- and *root* represents the root of the taxonomy in the knowledge graph, i.e. the node without ancestors $root = \{v \in V \mid \nexists (v,r,w) \in E_H\}$.

Thus, d meets all the properties of a metric:

- **Non-negativity**: The number of hops between two nodes cannot be negative: $\forall x,y, d(x,y) \geq 0$.
- **Identity**: The number of hops among a node and itself is $d(x,x) = 0$.
- **Symmetry**: The number of hops among two nodes is independent of the starting node: $d(x,y) = d(y,x)$.
- **Triangle inequality**: given that $d(x,y)$ represents the number of hops of the shortest path between x and y, it cannot exist another path with lower number of hops: $d(x,z) \leq d(x,y) + d(y,z)$.

Next, the metric properties are proved for d_{tax} and d_{ps} following a proof by contradiction strategy.

Proof of non-negativity If $D_{\text{tax}}(x,y)$ and $D_{\text{ps}}(x,y)$ are negative, then the function $d(x,y)$ must be also negative, which is not true. The next two proofs show the contradictions:

Proof. Suppose that D_{tax} returns a negative value,

$$\begin{aligned} D_{\text{tax}}(x,y) &< 0 \\ \frac{d(lca(x,y),x) + d(lca(x,y),y)}{d(root,x) + d(root,y)} &< 0 \\ d(lca(x,y),x) + d(lca(x,y),y) &< 0 \end{aligned}$$

The distance in number of hops is always non-negative:

$$\forall a,b, d(a,b) \geq 0$$

Hence, a contradiction is reached:

$$0 + 0 < 0 \qquad \square$$

Proof. Suppose that D_{ps} returns a negative value,

$$\begin{aligned} D_{\text{ps}}(x,y) &< 0 \\ 1 - \frac{d(root,lca(x,y))}{d(root,lca(x,y)) + d(lca(x,y),x) + d(lca(x,y),y)} &< 0 \\ \frac{d(root,lca(x,y))}{d(root,lca(x,y)) + d(lca(x,y),y) + d(lca(x,y),y)} &> 1 \\ d(root,lca(x,y)) + d(lca(x,y),x) + d(lca(x,y),y) &< d(root,lca(x,y)) \\ d(lca(x,y),x) + d(lca(x,y),y) &< 0 \end{aligned}$$

The distance in number of hops is always non-negative:

$$\forall a,b, d(a,b) \geq 0$$

Hence, a contradiction is reached:

$$0 + 0 < 0 \qquad \square$$

As d is a non-negative function, the addition of two non-negative values cannot be negative, so $D_{\text{tax}}(x,y) \geq 0$ and $D_{\text{ps}}(x,y) \geq 0$.

Proof of identity If $D_{\text{tax}}(x,x) \neq 0$ and $D_{\text{ps}}(x,y) \neq 0$ would not fulfill the identity property, then the function $d(x,y)$ would neither have it, which is not true. The following two proofs show the contradictions:

Proof. Suppose that $D_{\text{tax}}(x,x)$ is not 0,

$$\begin{aligned} D_{\text{tax}}(x,x) &\neq 0 \\ \frac{d(lca(x,x),x)+d(lca(x,x),x)}{d(root,x)+d(root,x)} &\neq 0 \\ d(lca(x,x),x)+d(lca(x,x),x) &\neq 0 \end{aligned}$$

The lowest common ancestor of a node is itself:

$$lca(x,x) = x$$

The function d fulfills the identity property

$$\begin{aligned} d(x,x) &= 0 \\ d(x,x)+d(x,x) &\neq 0 \\ 0 &\neq 0 \end{aligned}$$

□

Proof. Suppose that D_{ps} returns a negative value,

$$\begin{aligned} D_{\text{ps}}(x,x) &\neq 0 \\ 1 - \frac{d(root,lca(x,x))}{d(root,lca(x,x))+d(lca(x,x),x)+d(lca(x,x),x)} &\neq 0 \\ \frac{d(root,lca(x,x))}{d(root,lca(x,x))+d(lca(x,x),x)+d(lca(x,x),x)} &\neq 1 \\ d(root,lca(x,x))+d(lca(x,x),x)+d(lca(x,x),x) &\neq d(root,lca(x,x)) \end{aligned}$$

The lowest common ancestor of a node is itself:

$$lca(x,x) = x$$

The function d fulfills the identity property

$$\begin{aligned} d(x,x) &= 0 \\ d(x,x)+d(x,x) &\neq 0 \\ 0 &\neq 0 \end{aligned}$$

□

As d satisfies the identity property D_{tax} and D_{ps} must also meet it.

Proof of symmetry If $D_{\text{tax}}(x,y) \neq D_{\text{tax}}(y,x)$ and $D_{\text{ps}}(x,y) \neq D_{\text{ps}}(y,x)$ would not be symmetric, then either the function $d(x,y)$ or $lca(x,y)$ could not meet the symmetry property, which is not true. The following two proofs show the contradiction:

Proof. Suppose that D_{tax} is not symmetric and let $lca(x,y) = w$,

$$\begin{aligned} D_{\text{tax}}(x,y) &\neq D_{\text{tax}}(y,x) \\ \frac{d(lca(x,y),x)+d(lca(x,y),y)}{d(root,x)+d(root,y)} &\neq \frac{d(lca(y,x),y)+d(lca(y,x),x)}{d(root,y)+d(root,x)} \\ d(lca(x,y),x)+d(lca(x,y),y) &\neq d(lca(y,x),y)+d(lca(y,x),x) \\ d(w,x)+d(w,y) &\neq d(w,y)+d(w,x) \\ 0 &\neq 0 \end{aligned}$$

□

Proof. Suppose that D_{ps} is not symmetric and $lca(x,y) = w$,

$$\begin{aligned} D_{\text{ps}}(x,y) &\neq D_{\text{ps}}(y,x) \\ 1-\frac{d(root,w)}{d(root,w)+d(w,x)+d(w,y)} &\neq 1-\frac{d(root,w)}{d(root,w)+d(w,y)+d(w,x)} \\ \frac{d(root,w)}{d(root,w)+d(w,x)+d(w,y)} &\neq \frac{d(root,w)}{d(root,w)+d(w,y)+d(w,x)} \\ 1 &\neq 1 \end{aligned}$$

□

Given that functions d and lca are symmetric D_{tax} and D_{ps} satisfy also the symmetry property.

Proof of triangle inequality If $D_{\text{tax}}(x,z)$ and $D_{\text{ps}}(x,z)$ do not meet the triangle inequality property, then the function $d(x,y)$ could not satisfy such property, which is not true. The next two proofs show the contradictions:

Proof. Suppose that D_{tax} does not fulfill the triangle inequality and let $r = root$, $w = lca(x,z)$, $v = lca(x,y)$ and $k = lca(y,z)$,

$$\begin{aligned} D_{\text{tax}}(x,z) &> D_{\text{tax}}(x,y)+D_{\text{tax}}(y,z) \\ \frac{d(w,x)+d(w,z)}{d(r,x)+d(r,z)} &> \frac{d(v,x)+d(v,y)}{d(r,x)+d(r,y)}+\frac{d(k,y)+d(k,z)}{d(r,y)+d(r,z)} \end{aligned}$$

Suppose that:

$$d(v,y) = 0 \wedge d(k,y) = 0$$

This implies that y is a common ancestor of x and z:

$$\begin{aligned} v = y &= k \\ \frac{d(w,x)+d(w,z)}{d(r,x)+d(r,z)} &> \frac{d(y,x)}{d(r,x)+d(r,y)}+\frac{d(y,z)}{d(r,y)+d(r,z)} \\ \frac{d(w,x)+d(w,z)}{d(r,x)+d(r,z)} &> \frac{d(r,y)(d(x,y)+d(z,y))+d(r,x)d(z,y)+d(r,z)d(y,x)}{(d(r,x)+d(r,y))(d(r,y)+d(r,z)} \end{aligned}$$

Because of the properties of the Lowest Common Ancestor:

$$w = lca(x,z) \iff \forall n \in V, d(w,x)+d(w,z) \leq d(n,x)+d(n,z)$$

Therefore, the following statement applies:

$$d(w,x)+d(w,z) \leq d(x,y)+d(y,z)$$

$$\frac{d(w,x)+d(w,z)}{d(r,x)+d(r,z)} > \frac{d(r,y)(d(w,x)+d(w,z))+d(r,x)d(w,z)+d(r,z)d(w,x)}{(d(r,x)+d(r,y))(d(r,y)+d(r,z))}$$
$$\frac{d(w,x)+d(w,z)}{d(r,x)+d(r,z)} > \frac{d(r,y)(d(w,x)+d(w,z))+d(w,z)+d(w,x)}{(d(r,x)+d(r,y))(d(r,y)+d(r,z))}$$
$$\frac{d(w,x)+d(w,z)}{d(r,x)+d(r,z)} > \frac{d(r,y)(d(w,x)+d(w,z))+d(w,z)+d(w,x)}{(d(r,x)+d(r,y))(d(r,y)+d(r,z))}$$
$$\frac{d(w,x)+d(w,z)}{d(r,x)+d(r,z)} > \frac{d(r,y)(d(w,x)+d(w,z))+d(w,z)+d(w,x)}{d(r,x)d(r,y)+d(r,x)d(r,z)+d(r,y)^2+d(r,y)d(r,z)}$$

$$\begin{array}{c} d(w,x)[d(r,x)d(r,y)+d(r,y)d(r,z)+ \\ d(r,x)d(r,z)+d(r,y)^2]+ \\ d(w,z)[d(r,x)d(r,y)+d(r,y)d(r,z)+ \\ d(r,x)d(r,z)+d(r,y)^2] \end{array} > \begin{array}{c} d(w,x)[d(r,x)d(r,y)+d(r,y)d(r,z)+ \\ d(r,x)d(r,z)+d(r,z)^2]+ \\ d(w,z)[d(r,x)d(r,y)+d(r,y)d(r,z)+ \\ d(r,x)d(r,z)+d(r,x)^2] \end{array}$$

$$d(w,x)d(r,y)^2+d(w,z)d(r,y)^2 > d(w,x)d(r,z)^2+d(w,z)d(r,x)^2$$

Given that y is a common ancestor of x and z:

$$d(r,y) \leq d(r,x) \wedge d(r,y) \leq d(r,z)$$
$$d(r,y)^2 \leq d(r,x)^2 \wedge d(r,y)^2 \leq d(r,z)^2$$
$$d(w,z)d(r,y)^2 \leq d(w,z)d(r,x)^2 \wedge d(w,x)d(r,y)^2 \leq d(w,x)d(r,z)^2$$

Hence, the following contradiction is found:

$$d(w,z)d(r,y)^2+d(w,x)d(r,y)^2 \leq d(w,z)d(r,x)^2+d(w,x)d(r,x)^2$$
$$d(w,x)d(r,y)^2+d(w,z)d(r,y)^2 > d(w,x)d(r,z)^2+d(w,z)d(r,x)^2$$

□

Proof. Suppose that D_{ps} does not fulfill the triangle inequality,

$$D_{\text{ps}}(x,z) > D_{\text{ps}}(x,y)+D_{\text{ps}}(y,z)$$
$$1-\frac{d(r,w)}{d(r,w)+d(w,x)+d(w,z)} > 2-\frac{d(r,v)}{d(r,v)+d(v,x)+d(v,y)}-\frac{d(r,k)}{d(r,k)+d(k,y)+d(k,z)}$$
$$1+\frac{d(r,w)}{d(r,w)+d(w,x)+d(w,z)} < \frac{d(r,v)}{d(r,v)+d(v,x)+d(v,y)}+\frac{d(r,k)}{d(r,k)+d(k,y)+d(k,z)}$$

Suppose that:

$$d(v,y)=0 \wedge d(k,y)=0$$

This implies that y is a common ancestor of x and z:

$$v=y=k$$
$$1+\frac{d(r,w)}{d(r,w)+d(w,x)+d(w,z)} < \frac{d(r,y)}{d(r,y)+d(y,x)}+\frac{d(r,y)}{d(r,y)+d(y,z)}$$
$$1+\frac{d(r,w)}{d(r,w)+d(w,x)+d(w,z)} < \frac{2d(r,y)^2+d(r,y)d(y,z)+d(r,y)d(y,x)}{d(r,y)^2+d(r,y)d(y,z)+d(r,y)d(y,x)+d(y,x)d(y,z)}$$
$$\frac{2d(r,w)+d(w,x)+d(w,z)}{d(r,w)+d(w,x)+d(w,z)} < \frac{2d(r,y)^2+d(r,y)d(y,z)+d(r,y)d(y,x)}{d(r,y)^2+d(r,y)d(y,z)+d(r,y)d(y,x)+d(y,x)d(y,z)}$$

$$
\begin{array}{l}
2d(r,w)[(r,y)^2 + d(r,y)d(y,z) + \\
\quad d(r,y)d(y,x) + d(y,x)d(y,z)] + \\
d(w,x)[d(r,y)^2 + d(r,y)d(y,z) + \\
\quad d(r,y)d(y,x) + d(y,x)d(y,z)] + \\
d(w,z)[d(r,y)^2 + d(r,y)d(y,z) + \\
\quad d(r,y)d(y,x) + d(y,x)d(y,z)]
\end{array}
<
\begin{array}{l}
d(r,w)[2d(r,y)^2 + \\
\quad d(r,y)d(y,z) + d(r,y)d(y,x)] \\
d(w,x)[2d(r,y)^2 + \\
\quad d(r,y)d(y,z) + d(r,y)d(y,x)] \\
d(w,z)[2d(r,y)^2 + \\
\quad d(r,y)d(y,z) + d(r,y)d(y,x)]
\end{array}
$$

$$
\begin{array}{l}
d(r,w)[d(r,y)d(y,z) + \\
\quad d(r,y)d(y,x) + 2d(y,x)d(y,z)] + \\
d(w,x)[-d(r,y)^2 + d(y,x)d(y,z)] + \\
\quad d(w,z)[-d(r,y)^2 + d(y,x)d(y,z)]
\end{array}
< 0
$$

Because of the definition of Lowest Common Ancestor, the following statement is satisfied:

$$d(r,w) \geq d(r,y) \wedge d(w,x) + d(w,z) \leq d(y,x) + d(y,z)$$

Therefore, replacing $d(r,w)$ by $d(r,y)$ and $a \cdot d(y,x) + b \cdot d(y,z)$ by $a \cdot d(w,x) + b \cdot d(w,z)$ does not affect the relation:

$$
\begin{array}{l}
d(r,y)[d(r,y)d(w,z) + \\
\quad d(r,y)d(w,x) + 2d(w,x)d(w,z)] + \\
d(w,x)[-d(r,y)^2 + d(w,x)d(w,z)] + \\
\quad d(w,z)[-d(r,y)^2 + d(w,x)d(w,z)]
\end{array}
< 0
$$

$$
\begin{array}{l}
d(r,y)^2 d(w,z) + d(r,y)^2 d(w,x) + \\
2d(r,y)d(w,x)d(w,z) - d(r,y)^2 d(w,x) + \\
\quad d(w,x)^2 d(w,z) - d(r,y)^2 d(w,z) + \\
\qquad d(w,x)d(w,z)^2
\end{array}
< 0
$$

$$
\begin{array}{l}
2d(r,y)d(w,x)d(w,z) + \\
\quad d(w,x)^2 d(w,z) + d(w,x)d(w,z)^2
\end{array}
< 0
$$

The function d is non-negative, so the contradiction is reached:

$$0 > 0 \qquad \square$$

Sim$_{neigh}$ Let $G = (V,E,L)$ be a knowledge graph, $\mathrm{Sim}_{\mathrm{neigh}}$ is defined as follows:

$$
\mathrm{Sim}_{\mathrm{neigh}}(x,y) = 1 - \frac{\sum_{p_x \in N_x} \sum_{p_i \in N_y \setminus N_x} 1 - \mathrm{Sim}_{\mathrm{pair}}(p_x, p_j)}{|N_x \cup N_y||N_x|} - \frac{\sum_{p_y \in N_y} \sum_{p_i \in N_x \setminus N_y} 1 - \mathrm{Sim}_{\mathrm{pair}}(p_i, p_y)}{|N_x \cup N_y||N_y|}
$$

where $x, y \in V$ are nodes in the graph, $p_i, p_j, p_x, p_y \in E$ are edges in the graph and $\mathrm{Sim}_{\mathrm{pair}}$ is defined as follows:

$$\mathrm{Sim}_{\mathrm{pair}}(p_x, p_y) = \mathrm{Sim}_{\mathrm{hier}}(l_x, l_y) \cdot \mathrm{Sim}_{\mathrm{hier}}(w_x, w_y)$$

Remember that an edge $p \in E$ is described as a triple (v,l,w), where $v,w \in V$ are nodes in V and $l \in L$ represents the label of the edge.

In order to prove that $\text{Sim}_{\text{neigh}}$ satisfies the properties of a metric, it is needed to prove these properties also for Sim_{pair}.

Proof of non-negativity First, the non-negativity is proved for Sim_{pair}:

Proof. Let $p_x, p_y \in E$ be two edges in the knowledge graph:

$$\text{Sim}_{\text{pair}}(p_x,p_y) > 1$$
$$\text{Sim}_{\text{hier}}(l_x,l_y) \cdot \text{Sim}_{\text{hier}}(w_x,w_y) > 1$$

However, previously the inequality $\text{Sim}_{\text{hier}} \leq 1$ was proved. Thus, a contradiction is reached:

$$\text{Sim}_{\text{hier}}(l_x,l_y) \leq 1 \wedge \text{Sim}_{\text{hier}}(w_x,w_y) \leq 1$$
$$\text{Sim}_{\text{hier}}(l_x,l_y) \cdot \text{Sim}_{\text{hier}}(w_x,w_y) \leq 1$$
$$1 > 1 \square$$

Next, the non-negativity of $\text{Sim}_{\text{neigh}}$ is proved.

Proof. Let $x,y \in V$ be two nodes in the knowledge graph:

$$\text{Sim}_{\text{neigh}}(x,y) > 1$$
$$1 - \frac{\sum_{p_x \in N_x} \sum_{p_i \in N_y \setminus N_x} 1 - \text{Sim}_{\text{pair}}(p_x,p_j)}{|N_x \cup N_y||N_x|} - \frac{\sum_{p_y \in N_y} \sum_{p_i \in N_x \setminus N_y} 1 - \text{Sim}_{\text{pair}}(p_i,p_y)}{|N_x \cup N_y||N_y|} > 1$$
$$- \frac{\sum_{p_x \in N_x} \sum_{p_i \in N_y \setminus N_x} 1 - \text{Sim}_{\text{pair}}(p_x,p_j)}{|N_x \cup N_y||N_x|} - \frac{\sum_{p_y \in N_y} \sum_{p_i \in N_x \setminus N_y} 1 - \text{Sim}_{\text{pair}}(p_i,p_y)}{|N_x \cup N_y||N_y|} > 0$$

$$-\frac{|N_y|\sum_{p_x\in N_x}\sum_{p_i\in N_y\setminus N_x}1-\mathrm{Sim}_{\mathrm{pair}}(p_x,p_j)}{|N_x\cup N_y||N_x||N_y|}-\frac{|N_x|\sum_{p_y\in N_y}\sum_{p_i\in N_x\setminus N_y}1-\mathrm{Sim}_{\mathrm{pair}}(p_i,p_y)}{|N_x\cup N_y||N_x||N_y|}>0$$

$$-|N_y|\sum_{p_x\in N_x}\sum_{p_i\in N_y\setminus N_x}(1-\mathrm{Sim}_{\mathrm{pair}}(p_x,p_j))-|N_x|\sum_{p_y\in N_y}\sum_{p_i\in N_x\setminus N_y}(1-\mathrm{Sim}_{\mathrm{pair}}(p_i,p_y))>0$$

$$|N_y|\sum_{p_x\in N_x}\sum_{p_i\in N_y\setminus N_x}(1-\mathrm{Sim}_{\mathrm{pair}}(p_x,p_j))+|N_x|\sum_{p_y\in N_y}\sum_{p_i\in N_x\setminus N_y}(1-\mathrm{Sim}_{\mathrm{pair}}(p_i,p_y))<0$$

Given that $\forall p_i,p_j\in E,\mathrm{Sim}_{\mathrm{pair}}(p_i,p_j)\leq 1$, then $1-\mathrm{Sim}_{\mathrm{pair}}(p_i,p_j)\geq 0$ for all $p_i,p_j\in E$. Thus, the sum of non-negative numbers cannot be negative, so a contradiction is reached.

$$|N_y|\sum_{p_x\in N_x}\sum_{p_i\in N_y\setminus N_x}(1-1)+|N_x|\sum_{p_y\in N_y}\sum_{p_i\in N_x\setminus N_y}(1-1)<0$$
$$|N_y|\sum_{p_x\in N_x}\sum_{p_i\in N_y\setminus N_x}0+|N_x|\sum_{p_y\in N_y}\sum_{p_i\in N_x\setminus N_y}0<0$$
$$0<0\square$$

Proof of identity First, the identity condition is proved for $\mathrm{Sim}_{\mathrm{pair}}$.

Proof. Let $p_x,p_y\in E$ be two edges in the knowledge graph:

$$\mathrm{Sim}_{\mathrm{pair}}(p_x,p_x)\neq 1$$
$$\mathrm{Sim}_{\mathrm{hier}}(l_x,l_x)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_x,w_x)\neq 1$$

$\mathrm{Sim}_{\mathrm{hier}}$ meets the identity property:

$$\mathrm{Sim}_{\mathrm{hier}}(l_x,l_x)=1$$
$$\mathrm{Sim}_{\mathrm{hier}}(w_x,w_x)=1$$
$$\mathrm{Sim}_{\mathrm{hier}}(l_x,l_x)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_x,w_x)\neq 1$$
$$1\cdot 1\neq 1$$
$$1\neq 1\square$$

Next, the identity property is also proved for $\mathrm{Sim}_{\mathrm{neigh}}$.

Proof. Let $x\in V$ be a node in the knowledge graph:

$$\mathrm{Sim}_{\mathrm{neigh}}(x,x)\neq 1$$
$$1-\frac{\sum_{p_x\in N_x}\sum_{p_i\in N_x\setminus N_x}1-\mathrm{Sim}_{\mathrm{pair}}(p_x,p_j)}{|N_x\cup N_x||N_x|}-\frac{\sum_{p_x\in N_x}\sum_{p_i\in N_x\setminus N_x}1-\mathrm{Sim}_{\mathrm{pair}}(p_i,p_x)}{|N_x\cup N_x||N_x|}\neq 1$$
$$-\frac{\sum_{p_x\in N_x}\sum_{p_i\in N_x\setminus N_x}1-\mathrm{Sim}_{\mathrm{pair}}(p_x,p_j)}{|N_x\cup N_x||N_x|}-\frac{\sum_{p_x\in N_x}\sum_{p_i\in N_x\setminus N_x}1-\mathrm{Sim}_{\mathrm{pair}}(p_i,p_x)}{|N_x\cup N_x||N_x|}\neq 0$$

Given that $N_x \setminus N_x = \emptyset$, the summation is empty, and therefore the result is 0.

$$-\frac{\sum_{p_x \in N_x} \sum_{p_i \in \emptyset} 1 - \text{Sim}_{\text{pair}}(p_x, p_j)}{|N_x \cup N_x||N_x|} - \frac{\sum_{p_x \in N_x} \sum_{p_i \in \emptyset} 1 - \text{Sim}_{\text{pair}}(p_i, p_x)}{|N_x \cup N_x||N_x|} \neq 0$$

$$0 \neq 0 \square$$

Proof of symmetry First, the symmetry property is proved for Sim_{pair}.

Proof. Let $p_x, p_y \in E$ be two edges in the knowledge graph:

$$\text{Sim}_{\text{pair}}(p_x, p_y) \neq \text{Sim}_{\text{pair}}(p_y, p_x)$$
$$\text{Sim}_{\text{hier}}(l_x, l_y) \cdot \text{Sim}_{\text{hier}}(w_x, w_y) \neq \text{Sim}_{\text{hier}}(l_y, l_x) \cdot \text{Sim}_{\text{hier}}(w_y, w_x)$$

Sim_{hier} is symmetric:

$$\text{Sim}_{\text{hier}}(l_x, l_y) = \text{Sim}_{\text{hier}}(l_y, l_x)$$
$$\text{Sim}_{\text{hier}}(p_x, p_y) = \text{Sim}_{\text{hier}}(p_y, p_x)$$

Thus, a contradiction is reached:

$$\text{Sim}_{\text{hier}}(l_x, l_y) \cdot \text{Sim}_{\text{hier}}(w_x, w_y) \neq \text{Sim}_{\text{hier}}(l_y, l_x) \cdot \text{Sim}_{\text{hier}}(w_y, w_x) \square$$

Following, the symmetry property is demonstrated for $\text{Sim}_{\text{neigh}}$.

Proof. Let $p_x, p_y \in E$ be two edges in the knowledge graph:

$$\text{Sim}_{\text{neigh}}(x, y) \neq \text{Sim}_{\text{neigh}}(y, x)$$
$$1 - \frac{\sum_{p_x \in N_x} \sum_{p_i \in N_y \setminus N_x} 1 - \text{Sim}_{\text{pair}}(p_x, p_j)}{|N_x \cup N_y||N_x|} - \frac{\sum_{p_y \in N_y} \sum_{p_i \in N_x \setminus N_y} 1 - \text{Sim}_{\text{pair}}(p_i, p_y)}{|N_x \cup N_y||N_x|} \neq 1 - \frac{\sum_{p_y \in N_y} \sum_{p_i \in N_x \setminus N_y} 1 - \text{Sim}_{\text{pair}}(p_y, p_j)}{|N_y \cup N_x||N_y|} - \frac{\sum_{p_x \in N_x} \sum_{p_i \in N_y \setminus N_x} 1 - \text{Sim}_{\text{pair}}(p_i, p_x)}{|N_x \cup N_y||N_x|}$$
$$0 \neq 0 \square$$

Proof of triangle inequality First, the triangle inequality is proved for Sim_{pair}.

Proof. Let $p_x, p_y \in E$ be two edges in the knowledge graph:

$$\text{Sim}_{\text{pair}}(p_x, p_z) < \text{Sim}_{\text{pair}}(p_x, p_y) + \text{Sim}_{\text{pair}}(p_y, p_z) - 1$$
$$\text{Sim}_{\text{hier}}(l_x, l_z) \cdot \text{Sim}_{\text{hier}}(w_x, w_z) < \begin{array}{r} \text{Sim}_{\text{hier}}(l_x, l_y) \cdot \text{Sim}_{\text{hier}}(w_x, w_y) \\ + \text{Sim}_{\text{hier}}(l_y, l_z) \cdot \text{Sim}_{\text{hier}}(w_y, w_z) - 1 \end{array}$$

Sim_{hier} meets the triangle inequality property.

$$\text{Sim}_{\text{hier}}(l_x, l_z) \geq \text{Sim}_{\text{hier}}(l_x, l_y) + \text{Sim}_{\text{hier}}(l_y, l_z) - 1$$
$$\text{Sim}_{\text{hier}}(w_x, w_z) \geq \text{Sim}_{\text{hier}}(w_x, w_y) + \text{Sim}_{\text{hier}}(w_y, w_z) - 1$$

Multiplying the two equations:

$$\begin{aligned}\mathrm{Sim}_{\mathrm{hier}}(l_x,l_z)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_x,w_z)\geq\ & \mathrm{Sim}_{\mathrm{hier}}(l_x,l_y)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)+\\ & \mathrm{Sim}_{\mathrm{hier}}(l_x,l_y)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)+\\ & \mathrm{Sim}_{\mathrm{hier}}(l_y,l_z)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)+\\ & \mathrm{Sim}_{\mathrm{hier}}(l_y,l_z)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)-\\ & \mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)-\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)+1\end{aligned}$$

Because of the definition of $\mathrm{Sim}_{\mathrm{pair}}$:

$$\begin{aligned}\mathrm{Sim}_{\mathrm{pair}}(p_x,p_z)\geq\ & \mathrm{Sim}_{\mathrm{pair}}(p_x,p_y)+\mathrm{Sim}_{\mathrm{pair}}(p_y,p_z)\\ & \mathrm{Sim}_{\mathrm{hier}}(l_x,l_y)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)+\\ & \mathrm{Sim}_{\mathrm{hier}}(l_y,l_z)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)+\\ & -\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)-\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)+1\end{aligned}$$

Thus, the following substitution can be done:

$$\begin{aligned}&\mathrm{Sim}_{\mathrm{pair}}(p_x,p_y)+\mathrm{Sim}_{\mathrm{pair}}(p_y,p_z)\\ &\quad\mathrm{Sim}_{\mathrm{hier}}(l_x,l_y)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)+\\ &\quad\mathrm{Sim}_{\mathrm{hier}}(l_y,l_z)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)+\\ &\quad-\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)-\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)+1\end{aligned}\quad<\mathrm{Sim}_{\mathrm{pair}}(p_x,p_y)+\mathrm{Sim}_{\mathrm{pair}}(p_y,p_z)-1$$

$$\begin{aligned}&\mathrm{Sim}_{\mathrm{hier}}(l_x,l_y)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)+\\ &\quad\mathrm{Sim}_{\mathrm{hier}}(l_y,l_z)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)+\\ &\quad-\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)-\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)+1\end{aligned}\quad<-1$$

$$\begin{aligned}&\mathrm{Sim}_{\mathrm{hier}}(l_x,l_y)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)+\\ &\quad\mathrm{Sim}_{\mathrm{hier}}(l_y,l_z)\cdot\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)+\\ &\quad-\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)-\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)\end{aligned}\quad<-2$$

$$\begin{aligned}&\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)\cdot(\mathrm{Sim}_{\mathrm{hier}}(l_x,l_y)-1)+\\ &\quad\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)\cdot(\mathrm{Sim}_{\mathrm{hier}}(l_y,l_z)-1)\end{aligned}\quad<-2$$

$\mathrm{Sim}_{\mathrm{hier}}\in[0,1]$ values fluctuate between 0 and 1. In order to obtain the minimum value of the expression, positive values are minimized and negative values are maximized.

$$\begin{aligned}-\mathrm{Sim}_{\mathrm{hier}}(w_y,w_z)-\mathrm{Sim}_{\mathrm{hier}}(w_x,w_y)&<-2\\ -1-1&<-2\\ -2&<-2\square\end{aligned}$$

Let $s(X,Y)=\sum_{p_x\in X}\sum_{p_y\in Y}(1-\mathrm{Sim}_{\mathrm{pair}}(p_x,p_y))$ and $t(X,Y,Z)=|Z|s(X,Y)+|X|s(Y,Z)-|Y|s(X,Z)$. If $1-\mathrm{Sim}_{\mathrm{pair}}(p_x,p_y)$ meets the triangle inequality, then t also fulfills the property:

Proof. Let X,Y,Z be set of edges in a knowledge graph and let $d(p_x,p_y) = 1 - \mathrm{Sim}_{\mathrm{pair}}(p_x,p_y)$.

$$
\begin{aligned}
d(p_x,p_y) + d(p_y,p_z) - d(p_x,p_z) &\geq 0 \\
\sum_{p_x\in X}\sum_{p_y\in Y}\sum_{p_z\in Z} \frac{d(p_x,p_y) + d(p_y,p_z) - d(p_x,p_z)}{|X||Y||Z|} &\geq 0 \\
\frac{|Z|\sum_{p_x\in X}\sum_{p_y\in Y} d(p_x,p_y) + |X|\sum_{p_y\in Y}\sum_{p_z\in Z} d(p_y,p_z) - |Y|\sum_{p_x\in X}\sum_{p_z\in Z} d(p_x,p_z)}{|X||Y||Z|} &\geq 0 \\
|Z|\sum_{p_x\in X}\sum_{p_y\in Y} d(p_x,p_y) + |X|\sum_{p_y\in Y}\sum_{p_z\in Z} d(p_y,p_z) - |Y|\sum_{p_x\in X}\sum_{p_z\in Z} d(p_x,p_z) &\geq 0 \\
|Z|s(X,Y) + |X|s(Y,Z) - |Y|s(X,Z) &\geq 0 \\
t(X,Y,Z) &\geq 0 \square
\end{aligned}
$$

Next, the triangle inequality is proved for $\mathrm{Sim}_{\mathrm{neigh}}$. For that, the proof of Fujita [25] is followed.

Proof. Let $x,y,z \in V$ be nodes in the knowledge graph. Let also $s(X,Y) = \sum_{p_x\in N_x}\sum_{p_i\in N_y}(1 - \mathrm{Sim}_{\mathrm{pair}}(p_x,p_j))$

$$
1 - \frac{s(N_x,N_z\setminus N_x)}{|N_x\cup N_z||N_x|} - \frac{s(N_z,N_x\setminus N_z)}{|N_x\cup N_z||N_z|} < \begin{array}{l} 1 - \frac{s(N_x,N_y\setminus N_x)}{|N_x\cup N_y||N_x|} - \frac{s(N_y,N_x\setminus N_y)}{|N_x\cup N_z||N_y|} + \\ 1 - \frac{s(N_y,N_z\setminus N_y)}{|N_y\cup N_z||N_y|} - \frac{s(N_z,N_y\setminus N_z)}{|N_y\cup N_z||N_z|} - 1 \end{array}
$$

$$
-\frac{s(N_x,N_z\setminus N_x)}{|N_x\cup N_z||N_x|} - \frac{s(N_z,N_x\setminus N_z)}{|N_x\cup N_z||N_z|} < \begin{array}{l} -\frac{s(N_x,N_y\setminus N_x)}{|N_x\cup N_y||N_x|} - \frac{s(N_y,N_x\setminus N_y)}{|N_x\cup N_z||N_y|} \\ -\frac{s(N_y,N_z\setminus N_y)}{|N_y\cup N_z||N_y|} - \frac{s(N_z,N_y\setminus N_z)}{|N_y\cup N_z||N_z|} \end{array}
$$

$$
\frac{s(N_x,N_z\setminus N_x)}{|N_x\cup N_z||N_x|} + \frac{s(N_z,N_x\setminus N_z)}{|N_x\cup N_z||N_z|} > \begin{array}{l} \frac{s(N_x,N_y\setminus N_x)}{|N_x\cup N_y||N_x|} + \frac{s(N_y,N_x\setminus N_y)}{|N_x\cup N_z||N_y|} \\ + \frac{s(N_y,N_z\setminus N_y)}{|N_y\cup N_z||N_y|} + \frac{s(N_z,N_y\setminus N_z)}{|N_y\cup N_z||N_z|} \end{array}
$$

Let $N_x\cup N_y\cup N_z$ be partitioned into the following seven disjoint parts:

$$
\alpha = N_x\setminus(N_y\cup N_z), \quad \beta = N_y\setminus(N_x\cup N_z), \quad \gamma = N_z\setminus(N_x\cup N_y),
$$
$$
\delta = N_x\cap N_y\setminus N_z, \quad \varepsilon = N_y\cap N_z\setminus N_x, \quad \zeta = N_z\cap N_y\setminus N_y, \quad \eta = N_x\cap N_y\cap N_z,
$$

Let also $\theta = B\setminus\beta$. Then:

$$
\begin{aligned}
A &= \alpha\cup\delta\cup\zeta\cup\eta, & |A| &= |\alpha|+|\delta|+|\zeta|+|\eta|, \\
B &= \beta\cup\delta\cup\varepsilon\cup\eta, & |B| &= |\beta|+|\delta|+|\varepsilon|+|\eta|, \\
C &= \gamma\cup\varepsilon\cup\zeta\cup\eta, & |C| &= |\gamma|+|\varepsilon|+|\zeta|+|\eta|, \\
\theta &= \delta\cup\varepsilon\cup\eta, & |\theta| &= |\delta|+|\varepsilon|+|\eta|.
\end{aligned}
$$

$$|N_y||N_z||N_y \cup N_z||N_x \cup N_z|s(N_x, N_y \setminus N_x) + |N_x||N_z||N_y \cup N_z||N_x \cup N_z|s(N_y, N_x \setminus N_y) +$$
$$|N_x||N_z||N_x \cup N_y||N_x \cup N_z|s(N_y, N_z \setminus N_y) + |N_x||N_y||N_x \cup N_y||N_x \cup N_z|s(N_z, N_y \setminus N_z) -$$
$$|N_y||N_z||N_x \cup N_y||N_y \cup N_z|s(N_x, N_z \setminus N_x) - |N_x||N_y||N_x \cup N_y||N_y \cup N_z|s(N_z, N_x \setminus N_z) < 0$$

Applying the definitions of the disjoint sets:

$$|N_y||N_z| \cdot \Big[|\gamma||\delta \cup N_z|s(N_x, N_y \setminus N_x) + (|\gamma||\alpha| + |N_y \cup \zeta||C \setminus N_x|)s(N_x, \beta) +$$
$$|N_y \cup \zeta||N_x|(s(\beta, N_x \setminus N_z) + s(\beta, N_x \cap N_y)) + (|\beta||\gamma| + |N_y \cup N_z||N_x \cup \varepsilon|)s(A, \varepsilon)\Big] +$$
$$|N_x||N_z| \cdot \Big[(|\gamma||N_x \cup N_z| + |\zeta||\gamma| + |\zeta||N_x \cup \varepsilon|)s(\alpha, N_y) + |N_y||N_x \cup \varepsilon|(s(\alpha, N_y \setminus N_z) +$$
$$s(\alpha, N_y \cap N_z)) + |N_y||\gamma|(s(\alpha, \beta) + s(\alpha, \theta)) + [|\delta \cup N_z||\gamma| + |\delta \cup N_z||N_x \cup \varepsilon| +$$
$$|\beta||N_x \cup N_z|]s(\zeta, N_y)\Big] +$$
$$|N_x||N_y| \cdot \Big[|\alpha||N_x \cup \varepsilon|s(N_y \setminus N_z, N_z) + (\alpha||\gamma| + |\zeta \cup N_y||N_x \setminus N_z|)s(\beta, N_z) +$$
$$|\zeta \cup N_y||N_z|(s(\beta, N_x \cap N_z) + s(\beta, N_z \setminus N_x)) + (|\beta||\alpha| + |N_x \cup N_y||\delta \cup N_z|)s(\delta, N_z)\Big]$$
$$- |N_y||N_z| \cdot \Big[|\zeta \cup N_y||\beta|s(N_x, N_z \setminus N_x) + (|\alpha||\beta| + |N_y \setminus N_x||\delta \cup N_z|)s(N_x, \gamma) +$$
$$|N_x||\delta \cup N_z|(s(\alpha \cup \zeta, \gamma) + s(N_x \cap N_y, \gamma)\Big]$$
$$- |N_x||N_y| \cdot \Big[|\beta||N_y \cup \zeta|s(N_x \setminus N_z, N_z) + (|\beta||\gamma| + |N_x \cup \varepsilon||N_y \setminus N_z|)s(\alpha, N_z) +$$
$$|N_x \cup \varepsilon||N_z|(s(\alpha, \zeta \cup \gamma) + s(\alpha, N_y \cap N_z)\Big] + (|\alpha||\beta| + |N_x \cup N_y||\delta \cup N_z|)s(\delta, N_z)) < 0$$

$$|N_y||N_z| \cdot \Big[|\delta \cup N_z|t(N_x, N_y \setminus N_x, \gamma) + |\alpha|t(N_x, \beta, \gamma)\Big] + |N_y \cup \zeta|t(N_x, \beta, N_z \setminus N_x)) +$$
$$|N_x||N_y| \cdot \Big[|N_y \cup \varepsilon|t(\alpha, N_y \setminus N_z, N_z) + |\gamma|t(\alpha, \beta, N_z) + |N_y \cup \zeta|t(N_x \setminus N_z, \beta, N_z)\Big] +$$
$$|N_x||N_z| \cdot [(|N_x \cup N_z| + |\zeta|)t(\alpha, N_y, \gamma) + |N_y|t(\alpha, \theta, \gamma)] + |N_x||N_z| \cdot \Big[|N_x \cup \varepsilon|t(\alpha, N_y, \zeta) +$$
$$|N_z \cup \delta|t(\zeta, N_y, \gamma)\Big] + 2|N_x||N_z|(\delta \cup N_z||N_x \cup \varepsilon| + |\beta||N_x \cup N_z|)s(N_y, \zeta) +$$
$$2|N_x||N_y||N_z||N_y \cup \zeta|s(\beta, N_x \cap N_z) < 0$$

The result is a sum of non-negative values. Therefore, their sum has to be also non-negative, so a contradiction is reached:

$$0 < 0$$

□

Sim*shared* In this subsection the four properties of a metric are proved for $\mathrm{Sim}_{\mathrm{shared}}(x, y)$.

Let $x, y \in V$ be two nodes in a knowledge graph. $\mathrm{Sim}_{\mathrm{shared}}$ is defined as follows:

$$\mathrm{Sim}_{\mathrm{shared}}(x, y) = \frac{|\mathrm{Incident}(x) \cap \mathrm{Incident}(y)|}{|\mathrm{Incident}(x) \cup \mathrm{Incident}(y)|},$$

which is equivalent to the Jaccard coefficient of $\mathrm{Incident}(x)$ and $\mathrm{Incident}(y)$.

Proof of non-negativity By definition, the number of incident edges of a node has to be a non-negative number.

Proof. Let $x, y \in V$ be nodes in the knowledge graph:

$$\begin{aligned} \text{Sim}_{\text{shared}}(x,y) &> 1 \\ \frac{|\text{Incident}(x) \cap \text{Incident}(y)|}{|\text{Incident}(x) \cup \text{Incident}(y)|} &> 1 \\ |\text{Incident}(x) \cap \text{Incident}(y)| &> |\text{Incident}(x) \cup \text{Incident}(y)| \square \end{aligned}$$

Given that the size of the intersection of two sets cannot be bigger than the union, a contradiction is reached.

Proof of identity

Proof. Let x be a node in the knowledge graph:

$$\begin{aligned} \text{Sim}_{\text{shared}}(x,x) &\neq 1 \\ \frac{|\text{Incident}(x) \cap \text{Incident}(x)|}{|\text{Incident}(x) \cup \text{Incident}(x)|} &\neq 1 \\ \frac{|\text{Incident}(x)|}{|\text{Incident}(x)|} &\neq 1 \\ 1 &\neq 1 \square \end{aligned}$$

Proof of symmetry

Proof. Let $x, y \in V$ be nodes in the knowledge graph:

$$\begin{aligned} \text{Sim}_{\text{shared}}(x,y) &\neq \text{Sim}_{\text{shared}}(y,x) \\ \frac{|\text{Incident}(x) \cap \text{Incident}(y)|}{|\text{Incident}(x) \cup \text{Incident}(y)|} &\neq \frac{|\text{Incident}(y) \cap \text{Incident}(x)|}{|\text{Incident}(y) \cup \text{Incident}(x)|} \\ 1 &\neq 1 \square \end{aligned}$$

Proof of triangle inequality Let $x, y, z \in V$ be nodes in the knowledge graph. Let $X = \text{Incident}(x)$, $Y = \text{Incident}(y)$ and $Z = \text{Incident}(z)$ be the sets of incident edges of x, y and z respectively. In order to prove the triangle inequality for $\text{Sim}_{\text{shared}}$, the proof proposed by Levandowsky et al. [56] is followed. Thus, $X \cup Y \cup Z$ is divided into seven disjoint sets as in Figure A.1:

$$\begin{aligned} A &= X \setminus (Y \cup Z), \quad & B &= Y \setminus (A \cup Z), \quad & C &= Z \setminus (X \cup Y), \\ D &= X \setminus (A \cup Z), \quad & E &= Y \setminus (B \cup X), \quad & F &= Z \setminus (C \cup Y), \\ G &= X \setminus (A \cup D \cup F) \end{aligned}$$

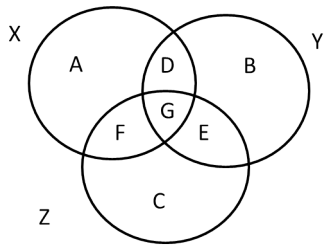


Figure A.1: Naming of sets and subsets in $X \cup Y \cup Z$.

Let a,b,c,d,e,f,g represent the sizes of the respective subsets. Let $v = a+b+c+d+e+f+g$. Next, the triangle inequality is proved for $\text{Sim}_{\text{shared}}$ by contradiction.

Proof. Let $x,y \in V$ be nodes in the knowledge graph:

$$
\begin{aligned}
\text{Sim}_{\text{shared}}(x,z) &< \text{Sim}_{\text{shared}}(x,y) + \text{Sim}_{\text{shared}}(y,z) - 1 \\
\frac{|X \cap Z|}{|X \cup Z|} &< \frac{|X \cap Y|}{|X \cup Y|} + \frac{|Y \cap Z|}{|Y \cup Z|} - 1 \\
\frac{f+g}{a+c+d+e+f+g} &< \frac{d+g}{a+b+d+e+f+g} + \frac{e+g}{b+c+d+e+f+g} - 1 \\
\frac{f+g}{v-b} &< \frac{d+g}{v-c} + \frac{e+g}{v-a} - 1 \\
\frac{f'}{v-b} &< \frac{d'}{v-c} + \frac{e'}{v-a} - 1 \\
\frac{f'}{v-b} &< \frac{(v-a)d' + (v-c)e' - (v-a)(v-c)}{(v-a)(v-c)} \\
(v-a)(v-c)f' &< \begin{aligned}[t] &(v-a)(v-b)d' + (v-b)(v-c)e' \\ &- (v-a)(v-b)(v-c)\end{aligned}
\end{aligned}
$$

$$
\begin{aligned}
f'(v^2 - av - cv + ac) < \; & (v^2 - av - bv + ab)d' + (v^2 - bv - cv + bc)e' \\
& - (v^3 - v^2a - v^2b - v^2c + \\
& \quad vab + vac + vbc - abc)
\end{aligned}
$$

$$
\begin{aligned}
0 < \; & -v^3 + v^2(a+b+c+d+e-f+g) \\
& + v(-ab - ac - bc - ad' - be' - bd' - ce' + af' + cf') \\
& + abc + abd' + bce' - acf'
\end{aligned}
$$

$a+b+c+d+e-f+g$ can be expressed as $v-2f$ and v^3 as v^2v.

$$
\begin{aligned}
0 < \; & v^2(v-2f-v) \\
& + v(-ab - ac - bc - ad' - be' - bd' - ce' + af' + cf') \\
& + abc + abd' + bce' - acf'
\end{aligned}
$$

$$
\begin{aligned}
0 < \; & v(-2fv) \\
& + v(-ab - ac - bc - ad' - be' - bd' - ce' + af' + cf') \\
& + abc + abd' + bce' - acf'
\end{aligned}
$$

$$
\begin{aligned}
0 < \; & v(-2fv - ab - ac - bc - ad' - be' - bd' - ce' + af' + cf') \\
& + abc + abd' + bce' - acf'
\end{aligned}
$$

$$
\begin{aligned}
0 < \; & v(-2fv - ab - ac - bc - ad - be - bd - 2bg - ce + af + cf) \\
& + abc + abd' + bce' - acf'
\end{aligned}
$$

Given that $2fv > af + cf$:

$$
\begin{aligned}
0 < \; & v(-ab - ac - bc - ad - be - bd - 2bg - ce) \\
& + abc + abd + abg + bce + bcg - acf - acg
\end{aligned}
$$

Because $v = a+b+c+d+e+f+g$, the positive terms outside the parenthesis cancel with negative terms inside.

$$0 < v(-be - bd - bg - ce) - acf - acg$$
$$0 < 0\square$$

Conclusion: Similarity-based machine learning approaches like K-Nearest Neighbors and K-Means require of distance or similarity measures that fulfill the metric properties in order to be efficient in terms of time and to deliver consistent results. In this section the metric properties are proved for the three defined components of GADES, i.e., Sim_{hier}, $\text{Sim}_{\text{neigh}}$ and $\text{Sim}_{\text{shared}}$. Whenever the GADES component are aggregated with a triangular norm, GADES will also fulfill the metric properties. Thus, GADES can be used in similarity-based machine learning algorithms and, therefore, semantics can be considered by these approaches.

List of Figures

List of Tables

Bibliography

[1] Agirre, E., Cer, D., Diab, M., and Gonzalez-Agirre, A. (2012). Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, volume 2, pages 385–393.

[2] Al-Rfou, R., Perozzi, B., and Skiena, S. (2013). Polyglot: Distributed word representations for multilingual NLP. *CoRR*, abs/1307.1662.

[3] Amdahl, G. M. (2013). Computer Architecture and Amdahl's Law. *IEEE Computer*, 46(12):38–46.

[4] Ashby, F. G. and Ennis, D. M. (2007). Similarity measures. *Scholarpedia*, 2(12):4116.

[5] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. G. (2007). Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007.*, pages 722–735.

[6] Baraty, S., Simovici, D. A., and Zara, C. (2011). The impact of triangular inequality violations on medoid-based clustering. In *Foundations of Intelligent Systems - Proceedings of 19th International Symposium on Methodologies for Intelligent Systems ISMIS 2011, Warsaw, Poland, June 28-30, 2011.*, pages 280–289.

[7] Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA*, volume 1, pages 238–247.

[8] Baumgartner, W. A., Cohen, K. B., Fox, L. M., Acquaah-Mensah, G. K., and Hunter, L. (2007). Manual curation is not sufficient for annotation of genomic databases. In *Proceedings 15th International Conference on Intelligent Systems for Molecular Biology (ISMB) & 6th European Conference on Computational Biology (ECCB), Vienna, Austria, July 21-25, 2007*, pages 41–48.

[9] Belanche, L. and Orozco, J. (2011). Things to know about a (dis)similarity measure. In *Proceedings of 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES) 2011, Kaiserslautern, Germany, September 12-14, 2011*, volume 1, pages 100–109.

[10] Benik, J., Chang, C., Raschid, L., Vidal, M., Palma, G., and Thor, A. (2012). Finding cross genome patterns in annotation graphs. In *Proceedings of 8th International Conference on Data Integration in the Life Sciences (DILS) 2012, College Park, MD, USA, June 28-29, 2012*, pages 21–36.

[11] Brickley, D. and Guha, R. (2014). RDF schema 1.1. W3C recommendation, W3C. http://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

[12] Buscaldi, D., Tournier, R., Aussenac-Gilles, N., and Mothe, J. (2012). IRIT: textual similarity combining conceptual similarity with an n-gram comparison method. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 552–556.

[13] Collarana, D., Galkin, M., Traverso-Ribón, I., Lange, C., Vidal, M., and Auer, S. (2017a). Semantic data integration for knowledge graph construction at query time. In *Proceedings of 11th IEEE International Conference on Semantic Computing, ICSC 2017, San Diego, CA, USA, January 30 - February 1, 2017*, pages 109–116.

[14] Collarana, D., Galkin, M., Traverso-Ribón, I., Vidal, M., Lange, C., and Auer, S. (2017b). Minte: Semantically integrating rdf graphs. In *Proceedings of 7th ACM International Conference on Web Intelligence, Mining and Semantics, WIMS 2017, Amantea, Italy, June 19 - 22, 2017*.

[15] Couto, F. M. and Silva, M. J. (2011). Disjunctive shared information between ontology concepts: application to gene ontology. *Journal of Biomedical Semantics*, 2:5.

[16] Couto, F. M., Silva, M. J., and Coutinho, P. (2007). Measuring semantic similarity between gene ontology terms. *Data Knowledge Engineering*, 61(1):137–152.

[17] Devos, D. and Valencia, A. (2000). Practical limits of function prediction. *Proteins: Structure, Function, and Bioinformatics*, 41(1):98–107.

[18] Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., and Vapnik, V. (1996). Support vector regression machines. In *Advances in Neural Information Processing Systems 9, (NIPS), Denver, CO, USA, December 2-5, 1996*, pages 155–161.

[19] Dürst, M. and Suignard, M. (2005). Internationalized resource identifiers (iris). Technical report, IEFT. https://www.ietf.org/rfc/rfc3987.txt.

[20] Epasto, A., Lattanzi, S., Mirrokni, V. S., Sebe, I., Taei, A., and Verma, S. (2015). Ego-net community mining applied to friend suggestion. *Proceedings of the Very Large Data Bases VLDB Endowment*, 9(4):324–335.

[21] Fernández, J. D., Llaves, A., and Corcho, Ó. (2014). Efficient RDF interchange (ERI) format for RDF data streams. In *Proceedings of 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 19-23, 2014*, volume 2, pages 244–259.

[22] Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., Potter, S. C., Punta, M., Qureshi, M., Sangrador-Vegas, A., Salazar, G. A., Tate, J. G., and Bateman, A. (2016). The pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, 44(1):279–285.

[23] Fischer, P. M., Lausen, G., Schätzle, A., and Schmidt, M. (2015). RDF constraint checking. In *Proceedings of International Conference on Extending Database Technology and International Conference on Database Theory 2015 Joint Conference.*, pages 205–212.

[24] Flores, A., Vidal, M., and Palma, G. (2015). Exploiting semantics to predict potential novel links from dense subgraphs. In *Proceedings of the 9th Alberto Mendelzon International Workshop on Foundations of Data Management, Lima, Peru, May 6 - 8, 2015.*

[25] Fujita, O. (2013). Metrics based on average distance between sets. *Japan Journal of Industrial and Applied Mathematics*, 30(1):1–19.

[26] Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, January 6-12, 2007*, pages 1606–1611.

[27] Galkin, M., Collarana, D., Traverso-Ribon, I., Vidal, M.-E., and Auer, S. (2017). Sjoin: A semantic join operator to integrate heterogeneous rdf graphs. In *Proceedings of 28th International Conference on Database and Expert Systems Applications (DEXA), Lyon, France*.

[28] Gene Ontology Consortium (2015). Gene ontology consortium: going forward. *Nucleic Acids Research*, 43(1):1049–1056.

[29] Glimm, B., Horrocks, I., Motik, B., Stoilos, G., and Wang, Z. (2014). Hermit: An OWL 2 Reasoner. *Journal of Automated Reasoning*, 53(3):245–269.

[30] Groß, A., Hartung, M., Kirsten, T., and Rahm, E. (2009). Estimating the quality of ontology-based annotations by considering evolutionary changes. In *Proceedings of 6th International Workshop on Data Integration in the Life Sciences (DILS) 2009, Manchester, UK, July 20-22, 2009*, pages 71–87.

[31] Guha, R. V., Brickley, D., and Macbeth, S. (2016). Schema.org: evolution of structured data on the web. *Communications of ACM*, 59(2):44–51.

[32] Hall, M. A., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18.

[33] Han, Y. (2007). Computing lowest common ancestors in directed acyclic graphs. In *Proceedings of 22nd International Conference on Computers and Their Applications, CATA-2007, Honolulu, Hawaii, USA, March 28-30, 2007*, pages 36–37.

[34] Harris, S. and Seaborne, A. (2013). SPARQL 1.1 query language. W3C recommendation, W3C. http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[35] Hassan, S. and Mihalcea, R. (2011). Semantic relatedness using salient semantic analysis. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*.

[36] He, Q., Chen, B.-C., and Argawal, D. (2016). Building the linkedin knowledge graph. `https://engineering.linkedin.com/blog/2016/10/building-the-linkedin-knowledge-graph`.

[37] Horst, H. J. (2005). Completeness, decidability and complexity of entailment for RDF schema and a semantic extension involving the OWL vocabulary. *Journal of Web Semantics*, 3(2-3):79–115.

[38] Huntley, R. P., Sawford, T., Mutowo-Meullenet, P., Shypitsyna, A., Bonilla, C., Martin, M. J., and O'Donovan, C. (2015). The GOA database: Gene ontology annotation updates for 2015. *Nucleic Acids Research*, 43(1):1057–1063.

[39] Isele, R. and Bizer, C. (2013). Active learning of expressive linkage rules using genetic programming. *Journal of Web Semantics*, 23:2–15.

[40] Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420.

[41] Jeh, G. and Widom, J. (2002). Simrank: a measure of structural-context similarity. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 538–543.

[42] Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of ROCLING X, Taiwan, 1997*.

[43] Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392.

[44] Kastrin, A., Rindflesch, T. C., and Hristovski, D. (2014). Link prediction on the semantic MEDLINE network - an approach to literature-based discovery. In *Proceedings of - 17th International Conference on Discovery Science (DS) 2014, Bled, Slovenia, October 8-10, 2014.*, pages 135–143.

[45] Kazakov, Y. (2008). RIQ and SROIQ are harder than SHOIQ. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR) 2008, Sydney, Australia, September 16-19, 2008*, pages 274–284.

[46] Klement, E., Mesiar, R., and Pap, E. (2000). *Triangular Norms*. Trends in Logic. Springer Netherlands.

[47] Knap, T., Kukhar, M., Machác, B., Skoda, P., Tomes, J., and Vojt, J. (2014). UnifiedViews: An ETL framework for sustainable RDF data processing. In *The Semantic Web: Extended Semantic Web Conference 2014 Satellite Events - ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, pages 379–383.

[48] Knoblock, C. A., Szekely, P. A., Ambite, J. L., Goel, A., Gupta, S., Lerman, K., Muslea, M., Taheriyan, M., and Mallick, P. (2012). Semi-automatically mapping structured sources into the semantic web. In *Proceedings of 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012*, pages 375–390.

[49] Köhler, S., Doelken, S. C., Mungall, C. J., Bauer, S., Firth, H. V., Bailleul-Forestier, I., Black, G. C., Brown, D. L., Brudno, M., Campbell, J., et al. (2014). The human phenotype ontology project: linking molecular biology and disease through phenotype data. *Nucleic Acids Research*, 42(Database-Issue):966–974.

[50] Kuhn, H. W. (2010). The hungarian method for the assignment problem. In *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, pages 29–47. Springer.

[51] Landauer, T. K., Laham, D., Rehder, B., and Schreiner, M. E. (1997). How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. In *Proceedings of the 19th annual meeting of the Cognitive Science Society*, pages 412–417.

[52] Lausen, G., Meier, M., and Schmidt, M. (2008). Sparqling constraints for RDF. In *11th International Conference on Extending Database Technology (EDBT) 2008,, Nantes, France, March 25-29, 2008, Proceedings*, pages 499–509.

[53] Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning (ICML) 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196.

[54] Lee, M. D. and Welsh, M. (2005). An empirical evaluation of models of text document similarity. In *Proceedings of the Cognitive Science Society*, volume 27, pages 1254–1259. Erlbaum.

[55] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., et al. (2015). Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

[56] Levandowsky, M. and Winter, D. (1971). Distance between sets. *Nature*, 234(5323):34–35.

[57] Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031.

[58] Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML) 1998, Madison, Wisconsin, USA, July 24-27, 1998*, pages 296–304.

[59] Mahdisoltani, F., Biega, J., and Suchanek, F. M. (2015). YAGO3: A knowledge base from multilingual wikipedias. In *CIDR 2015, Seventh Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*.

[60] Mazandu, G. K. and Mulder, N. J. (2013). Information content-based gene ontology semantic similarity approaches: toward a unified framework theory. *BioMed research international*.

[61] McGuinness, D., Kendall, E., Patel-Schneider, P., and Bao, J. (2012). OWL 2 web ontology language quick reference guide (second edition). W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-quick-reference-20121211/.

[62] Menger, K. (1942). Statistical metrics. *Proceedings of the National Academy of Sciences*, 28(12):535–537.

[63] Michelfeit, J., Knap, T., and Necaský, M. (2014). Linked data integration with conflicts. *CoRR*, abs/1410.7990.

[64] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

[65] Motik, B., Wu, Z., Horrocks, I., Grau, B. C., and Fokoue, A. (2012). OWL 2 web ontology language profiles (second edition). W3C recommendation, W3C. http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/.

[66] NC-IUBMB (1965). *Enzyme nomenclature: recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the nomenclature and classification of enzymes*. Academic Press.

[67] Ngomo, A. N. and Auer, S. (2011). LIMES - A time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI) 2011, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2312–2317.

[68] Nunes, B. P., Dietze, S., Casanova, M. A., Kawase, R., Fetahu, B., and Nejdl, W. (2013a). Combining a co-occurrence-based and a semantic measure for entity linking. In *Proceedings of 10th Extended Semantic Web Conference (ESWC) 2013, Montpellier, France, May 26-30, 2013. Proceedings*, pages 548–562.

[69] Nunes, B. P., Fetahu, B., Dietze, S., and Casanova, M. A. (2013b). Cite4me: A semantic search and retrieval web application for scientific publications. In *Proceedings of the International Semantic Web Conference 2013 Posters & Demonstrations Track, Sydney, Australia, October 23, 2013*, pages 25–28.

[70] Palma, G., Vidal, M., Haag, E., Raschid, L., and Thor, A. (2013). Measuring relatedness between scientific entities in annotation datasets. In *ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics. ACM-BCB 2013, Washington, DC, USA, September 22-25, 2013*, page 367.

[71] Palma, G., Vidal, M., and Raschid, L. (2014). Drug-target interaction prediction using semantic similarity and edge partitioning. In *Proceedings of 13th International Semantic Web Conference (ISWC) 2014, Riva del Garda, Italy, October 19-23, 2014*, volume 1, pages 131–146.

[72] Paul, C., Rettinger, A., Mogadala, A., Knoblock, C. A., and Szekely, P. A. (2016). Efficient graph-based document similarity. In *Proceedings of 13th Extended Semantic Web Conference (ESWC) 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016*, pages 334–349.

[73] Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3):489–508.

[74] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

[75] Pekar, V. and Staab, S. (2002). Taxonomy learning - factoring the structure of a taxonomy into a semantic classification decision. In *Proceedings of 19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*.

[76] Pérez, J., Arenas, M., and Gutierrez, C. (2009). Semantics and complexity of SPARQL. *ACM Transactions on Database Systems*, 34(3):16:1–16:45.

[77] Pesquita, C., Faria, D., Bastos, H., Falcao, A., and Couto, F. (2007). Evaluating go-based semantic similarity measures. In *Proceedings 10th Annual Bio-Ontologies Meeting*, page 38.

[78] Pesquita, C., Faria, D., Bastos, H. P., Ferreira, A. E. N., Falcão, A. O., and Couto, F. M. (2008). Metrics for GO based protein semantic similarity: a systematic evaluation. *BMC Bioinformatics*, 9(S-5).

[79] Pesquita, C., Pessoa, D., Faria, D., and Couto, F. (2009). Cessm: Collaborative evaluation of semantic similarity measures. *Jornadas de Bioinformatica B2009: Challenges in Bioinformatics*, 157:190.

[80] Redondo-García, J. L., Sabatino, M., Lisena, P., and Troncy, R. (2014). Detecting hot spots in web videos. In *Proceedings of the 13th International Semantic Web Conference (ISWC) 2014, Posters & Demonstrations Track, Riva del Garda, Italy, October 21, 2014.*, pages 141–144.

[81] Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *2010 International Conference on Languages Resources and Evaluation Workshop on New Challenges for NLP Frameworks*, pages 46–50. ELRA.

[82] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, volume 1 of *IJCAI'95*, pages 448–453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[83] Resnik, P. (2011). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *CoRR*, abs/1105.5444.

[84] Rindflesch, T. C., Kilicoglu, H., Fiszman, M., Rosemblat, G., and Shin, D. (2011). Semantic medline: an advanced information management application for biomedicine. *Information Services and Use*, 31(1-2).

[85] Ross, S. (1976). *A First course in probability*. Macmillan.

[86] Sachan, M. and Ichise, R. (2010). Using semantic information to improve link prediction results in network datasets. *International Journal of Engineering and Technology*, 2(4):334.

[87] Sanderson, R., Ciccarese, P., Van de Sompel, H., Bradshaw, S., Brickley, D., Castro, L. J. G., Clark, T., Cole, T., Desenne, P., Gerber, A., et al. (2013). Open annotation data model. Technical report, W3C.

[88] Schuhmacher, M. and Ponzetto, S. P. (2014). Knowledge-based graph document modeling. In *Proceedings of 7th ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 543–552.

[89] Schultz, A., Matteini, A., Isele, R., Bizer, C., and Becker, C. (2011). LDIF - linked data integration framework. In *Proceedings of the Second International Conference on Consuming Linked Data*, volume 782, pages 125–130.

[90] Schwartz, J., Steger, A., and Weißl, A. (2005). Fast algorithms for weighted bipartite matching. In *Proceedings of 4th International Workshop Experimental and Efficient Algorithms, WEA 2005, Santorini Island, Greece, May 10-13, 2005*, pages 476–487.

[91] Sevilla, J. L., Segura, V., Podhorski, A., Guruceaga, E., Mato, J. M., Martínez-Cruz, L. A., Corrales, F. J., and Rubio, A. (2005). Correlation between gene expression and go semantic similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4):330–338.

[92] Shi, C., Kong, X., Huang, Y., Yu, P. S., and Wu, B. (2013). Hetesim: A general framework for relevance measure in heterogeneous networks. *CoRR*, abs/1309.7393.

[93] Singhal, A. (2012). Introducing the knowledge graph: things, not strings. *Official google blog*.

[94] Skunca, N., Altenhoff, A. M., and Dessimoz, C. (2012). Quality of computationally inferred gene ontology annotations. *PLoS Computational Biology*, 8(5).

[95] Steiner, T., Troncy, R., and Hausenblas, M. (2010). How google is using linked data today and vision for tomorrow. *Proceedings of Linked Data in the Future Internet*, 700.

[96] Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data Knowledge Engineering*, 25(1-2):161–197.

[97] Sun, Y., Han, J., Yan, X., Yu, P. S., and Wu, T. (2011). Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *Proceedings of the Very Large Data Bases VLDB Endowment*, 4(11):992–1003.

[98] Taibi, D., Chawla, S., Dietze, S., Marenzi, I., and Fetahu, B. (2015). Exploring ted talks as linked data for education. *British Journal of Educational Technology*, 46(5):1092–1096.

[99] Tanon, T. P., Vrandecic, D., Schaffert, S., Steiner, T., and Pintscher, L. (2016). From freebase to wikidata: The great migration. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 1419–1428.

[100] The UniProt Consortium (2017). Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, 45(1):158–169.

[101] The W3C SPARQL Working Group (2013). SPARQL 1.1 overview. W3C recommendation, W3C. http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/.

[102] Traverso-Ribón, I. (2015). Exploiting semantics from ontologies to enhance accuracy of similarity measures. In *Proceedings of 12th Extended Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015*, pages 795–805.

[103] Traverso-Ribón, I., Palma, G., Flores, A., and Vidal, M. (2016a). Considering semantics on the discovery of relations in knowledge graphs. In *Proceedings of 20th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2016, Bologna, Italy, November 19-23, 2016*, pages 666–680.

[104] Traverso-Ribón, I. and Vidal, M. (2015). Exploiting information content and semantics to accurately compute similarity of go-based annotated entities. In *Proceedings of IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB 2015, Niagara Falls, ON, Canada, August 12-15, 2015*, pages 1–8.

[105] Traverso-Ribón, I., Vidal, M., Kämpgen, B., and Sure-Vetter, Y. (2016b). GADES: A graph-based semantic similarity measure. In *Proceedings of the 12th International Conference on Semantic Systems, SEMANTICS 2016, Leipzig, Germany, September 12-15, 2016*, pages 101–104.

[106] Traverso-Ribón, I., Vidal, M., and Palma, G. (2015a). Annevol: An evolutionary framework to description ontology-based annotations. In *Proceedings of 11th International Conference on Data Integration in the Life Sciences, DILS 2015, Los Angeles, CA, USA, July 9-10, 2015*, pages 87–103.

[107] Traverso-Ribón, I., Vidal, M., and Palma, G. (2015b). Onsim: A similarity measure for determining relatedness between ontology terms. In *Proceedings of 11th International Conference on Data Integration in the Life Sciences, DILS 2015, Los Angeles, CA, USA, July 9-10, 2015*, pages 70–86.

[108] Tversky, A. (1977). Features of similarity. *Psychological review*, 84(4):327.

[109] Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

[110] Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research. Washington, DC*, pages 354–359.

[111] Wood, D., Lanthaler, M., and Cyganiak, R. (2014). RDF 1.1 concepts and abstract syntax. W3C recommendation, W3C. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[112] Zhou, Z., Qi, G., and Suntisrivaraporn, B. (2013). A new method of finding all justifications in owl 2 el. In *Proceedings of IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013*, volume 1, pages 213–220.

[113] Zhu, T. and Lan, M. (2012). Tiantianzhu7: System description of semantic textual similarity (STS) in the semeval-2012 (task 6). In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 575–578.