GARUM: A Semantic Similarity Measure Based on Machine Learning and Entity Characteristics

Ignacio Traverso-Ribón[1(✉)] and Maria-Esther Vidal[2,3]

[1] University of Cadiz, Cádiz, Spain
ignacio.traverso@uca.es
[2] L3S Research Center, Hanover, Germany
[3] TIB Leibniz Information Center for Science and Technology, Hanover, Germany
maria.vidal@tib.eu

Abstract. Knowledge graphs encode semantics that describes entities in terms of several *characteristics*, e.g., attributes, neighbors, class hierarchies, or association degrees. Several *data-driven* tasks, e.g., ranking, clustering, or link discovery, require for determining the relatedness between knowledge graph entities. However, state-of-the-art similarity measures may not consider all the characteristics of an entity to determine entity relatedness. We address the problem of similarity assessment between knowledge graph entities and devise GARUM, a semantic similarity measure for knowledge graphs. GARUM relies on similarities of entity characteristics and computes similarity values considering simultaneously several entity characteristics. This combination can be manually or automatically defined with the help of a machine learning approach. We empirically evaluate the accuracy of GARUM on knowledge graphs from different domains, e.g., networks of proteins and media news. In the experimental study, GARUM exhibits higher correlation with gold standards than studied existing approaches. Thus, these results suggest that similarity measures should not consider *entity characteristics* in isolation; contrary, combinations of these characteristics are required to precisely determine relatedness among entities in a knowledge graph. Further, the combination functions found by a machine learning approach outperform the results obtained by the manually defined aggregation functions.

1 Introduction

Semantic Web and Linked Data communities foster the publication of large volumes of data in the form of semantically annotated knowledge graphs. For example, knowledge graphs like DBpedia[1], Wikidata or Yago[2], represent general domain concepts such as musicians, actors, or sports, using RDF vocabularies.

[1] http://dbpedia.org.
[2] http://yago-knowledge.org.

S. Hartmann et al. (Eds.): DEXA 2018, LNCS 11029, pp. 169–183, 2018.
https://doi.org/10.1007/978-3-319-98809-2_11

Additionally, domain specific communities like Life Sciences and the financial domain, have also enthusiastically supported the collaborative development of diverse ontologies and semantic vocabularies to enhance the description of knowledge graph entities and reduce the ambiguity in such descriptions, e.g., the Gene Ontology (GO) [2], the Human Phenotype Ontology (HPO) [10], or the Financial Industry Business Ontology (FIBO)[3]. Knowledge graphs encode semantics that describe entities in terms of several *entity characteristics*, e.g., class hierarchies, neighbors, attributes, and association degrees. During the last years, several semantic similarity measures for knowledge graph entities have been proposed, e.g., GBSS [15], HeteSim [22], and PathSim [24]. However, these measures do not consider all the *entity characteristics* represented in a knowledge graph at the same time in a aggregated fashion. The importance of precisely determining relatedness in data-driven tasks, e.g., knowledge discovery, and the increasing size of existing knowledge graphs, introduce the challenge of defining semantic similarity measures able to exploit all the information described in knowledge graphs, i.e., all the *characteristics* of the represented entities.

We present GARUM, a GrAph entity Regression sUpported similarity Measure. GARUM exploits knowledge encoded in *characteristics* of an entity, i.e., ancestors or *hierarchies*, neighborhoods, associations, or *shared information*, and literals or *attributes*. GARUM receives a knowledge graph and two entities to be compared. As a result, GARUM returns a similarity value that aggregates similarity values computed based on the different *entity characteristics*; a domain-dependent aggregation function α combines similarity values specific for each *entity characteristic*. The function α can be either manually defined or predicted by a regression machine learning approach. The intuition is that knowledge represented in *entity characteristics*, precisely describes entities and allows for determining more accurate similarity values.

We conduct an empirical study with the aim of analyzing the impact of considering *entity characteristics* in the accuracy of a similarity measure over a knowledge graph. GARUM is evaluated over entities of three different knowledge graphs: The first knowledge graph describes news articles annotated with DBpedia entities; and the other two graphs describe proteins annotated with the Gene Ontology. GARUM is compared with state-of-the-art similarity measures with the goal of determining if GARUM similarity values are more correlated to the gold standards. Our experimental results suggest that: (*i*) Considering all *entity characteristics* allow for computing more accurate similarity values; (*ii*) GARUM is able to outperform state-of-art approaches obtaining higher values of correlation; and (*iii*) Machine learning approaches are able to predict aggregation functions that outperform the manually functions defined by humans.

The remainder of this article is structured as follows: Sect. 2 motivates our approach using a subgraph from DBpedia. Section 3 describes GARUM and Sect. 4 summarizes experimental results. Related work is presented in Sect. 5, and finally, Sect. 6 concludes and give insights for future work.

[3] https://www.w3.org/community/fibo/.

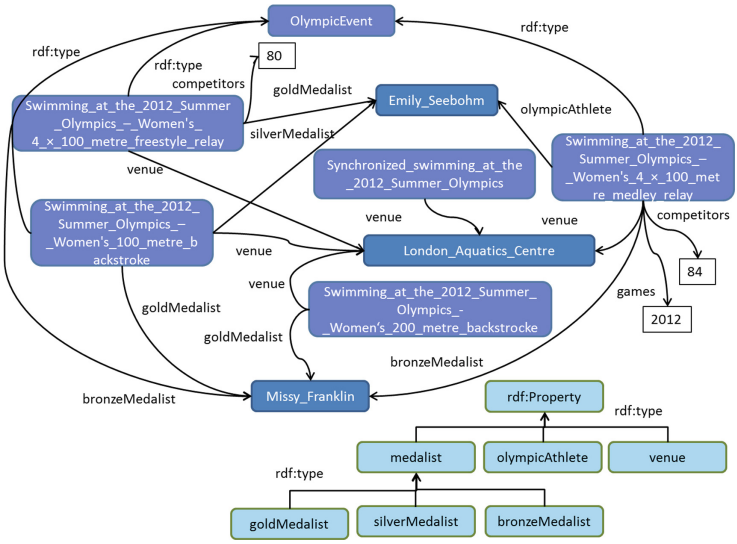


Fig. 1. Motivating Example. Two subgraphs from DBpedia. The above graph describes swimming events and entities related to these events, while the other graph represents a hierarchy of the properties in DBpedia.

2 Motivating Example

We motivate our work with a real-world knowledge graph extracted from DBpedia (Fig. 1); it describes swimming events in olympic games. Each event is related to other entities, e.g., athletes, locations, or years, using different relations or RDF *properties*, e.g., *goldMedalist* or *venue*. These RDF properties are also described in terms of the RDF property `rdf:type` as depicted in Fig. 1. Relatedness between entities is determined based on different *entity characteristics*, i.e., class hierarchy, neighbors, shared associations, and properties.

Consider entities *Swimming at the 2012 Summer Olympics - Women's 100 m backstroke*, *Swimming at the 2012 Summer Olympics - Women's 4x100 m freestyle relay*, and *Swimming at the 2012 Summer Olympics - Women's 4x100 m medley relay*. For the sake of clarity we rename them as *Women's 100 m backstroke*, *Women's 4x100 m freestyle*, and *Women's 4x100 m medley relay*, respectively. The entity hierarchy is induced by the *rdf:type* property, which describes an entity as instance of an RDF class. Particularly, these swimming events are described as instances of the *OlympicEvent* class, which is at the fifth level of depth in the DBpedia ontology hierarchy. Thus, based on the knowledge encoded in this hierarchy, these entities are highly similar. Additionally, these entities share exactly the same set of neighbors that is formed by the entities *Emily Seebohm*, *Missy Franklin*, and *London Aquatic Centre*. However, the relations with *Emily Seebohm* and *Missy Franklin* are different. *Women's 4x100 m freestyle* and *Women's 100 m backstroke* are related with *Emily Seebohm* through properties

goldMedalist and *silverMedalist*, respectively, and with *Missy Franklin* through properties *bronzeMedalist* and *goldMedalist*. Nevertheless, *Women's 4x100 m medley relay* is related with *Missy Franklin* through the property *bronzeMedalist*, and with *Emily Seebohm* through *olympicAthlete*. Considering only the entities in these neighborhoods, they are identical since they share exactly the same set of neighbors. However, whenever properties labels and the property hierarchy are considered, we observe that *Women's 4x100 m freestyle* and *Women's 100 m backstroke* are more similar since in both events *Missy Franklin* and *Emily Seebohm* are *medalists*, while in *Women's 4x100 m medley relay* only *Missy Franklin* is *medalist*. Furthermore, swimming events are also related with attributes through datatype properties. For the sake of clarity, we only include a portion of these attributes in Fig. 1. Considering these attributes, 84 athletes participated in *Women's 4x100 m medley relay*, while only 80 participated in *Women's 4x100 m freestyle*. Finally, the node degree or *shared information* is different for each entity in the graph. Entities with a high node degree are considered abstract entities, while others with low node degree are considered specific. For instance, in Fig. 1, the entity *London Aquatic Centre* has five incident edges, while *Emily Seebohm* has four edges and *Missy Franklin* has only three incident edges. Thus, the entity *London Aquatic Centre* is less specific than *Emily Seebohm*, which is also less specific than *Missy Franklin*.

According to these observations, the similarity between two knowledge graph entities cannot be estimated only considering one *entity characteristic*. Hence, combinations of them may have to be taken into account to precisely determine relatedness between entities in a knowledge graph.

3 Our Approach: GARUM

We propose GARUM, a semantic similarity measure for determining relatedness between entities represented in knowledge graphs. GARUM considers the knowledge encoded in *entity characteristics*, e.g., hierarchies, neighborhoods, shared information, and attributes to accurately compute similarity values between entities in a knowledge graph. GARUM calculates values of similarity for each *entity characteristic* independently and combines these values to produce an aggregated similarity value between the compared entities. Figure 2 depicts the GARUM architecture. GARUM receives as input a knowledge graph G and two entities e_1, e_2 to be compared. *Entity characteristics* of the compared entities are extracted from the knowledge graph and compared as isolated elements.

Definition 1. *Knowledge graph. Given a set of entities V, a set of edges E, and a set of property labels L, a knowledge graph G is defined as $G = (V, E, L)$. An edge corresponds to a triple (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.*

Definition 2. *Individual similarity measure. Given a knowledge graph $G = (V, E, L)$, two entities e_1 and e_2 in V, and an entity characteristic $\mathcal{EC}$ of e_1 and e_2 in G, an individual similarity measure $Sim_{\mathcal{EC}}(e_1, e_2)$ corresponds to a similarity function defined in terms of $\mathcal{EC}$ for e_1 and e_2.*

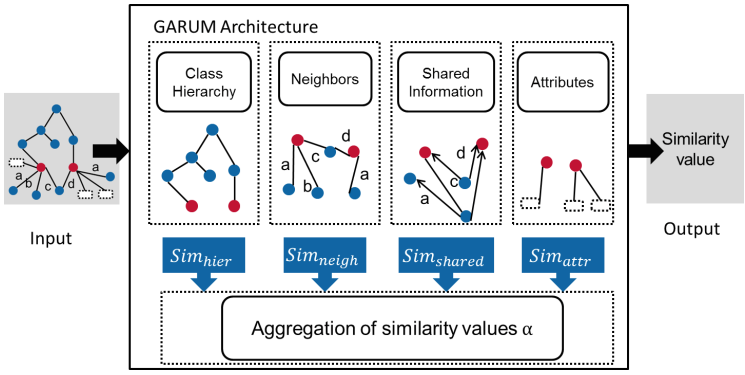


Fig. 2. The GARUM Architecture. GARUM receives a knowledge graph G and two entities to be compared (red nodes). Based on semantics encoded in the knowledge graph (blue nodes), GARUM computes similarity values in terms of class hierarchies, neighborhoods, shared information and the attributes of the input entities. Generated similarity values, $Sim_{hier}, Sim_{neigh}, Sim_{shared}, Sim_{attr}$, are combined using a function α. The aggregated value is returned as output. (Color figure online)

The hierarchical similarity $\text{Sim}_{\text{hier}}(e_1, e_2)$ or the neighborhood similarity $\text{Sim}_{\text{neigh}}(e_1, e_2)$ are examples of individual similarity measures. These individual similarity measures are combined using an aggregation function α. Next, we describe the four considered individual similarity measures.

Hierarchical Similarity: Given a knowledge graph G, a hierarchy is induced by a set of hierarchical edges $HE = \{(v_i, r, v_j) | (v_i, r, v_j) \in E \wedge \text{Hierarchical}(r)\}$. HE is a subset of edges in the knowledge graph whose property labels refer to a hierarchical relation, e.g., *rdf:type*, *rdfs:subClassOf*, or *skos:broader*. Generally, every relation that presents an entity as a generalization (ancestor) or an specification (successor) of another entity is a hierarchical relation. GARUM relies on existing hierarchical distance measures, e.g., d_{tax} [1] and d_{ps} [16] to determine the hierarchical similarity between entities; it is defined as follows:

$$\text{Sim}_{\text{hier}}(e_1, e_2) = \begin{cases} 1 - d_{\text{tax}}(e_1, e_2) \\ 1 - d_{\text{ps}}(e_1, e_2) \end{cases} \tag{1}$$

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., $N(e) = \{(r, e_i) | (e, r, e_i) \in E)$. With this definition of neighborhood, we can consider the neighbor entity and the relation type of the edge at the same time. GARUM 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 entity of the knowledge graph, regardless of it is an instance, a class or a relation. In order

to maximize the similarity between two neighborhoods, GARUM combines pair comparisons using the following formula:

$$\mathrm{Sim}_{\mathrm{neigh}}(e_1, e_2) = \frac{\sum_{i=0}^{|N(e_1)|} \max_{p_x \in N(e_2)} \mathrm{Sim}_{\mathrm{pair}}(p_i, p_x) + \sum_{j=0}^{|N(e_2)|} \max_{p_y \in N(e_1)} \mathrm{Sim}_{\mathrm{pair}}(p_j, p_y)}{|N(e_1)| + |N(e_2)|} \tag{2}$$

In Fig. 1, the neighborhoods of *Women's 100 m backstroke* and *Women's 4x100 m freestyle* are {(*venue*, *London Aquatic Centre*), (*silverMedalist*, *Emily Seebohm*), (*goldMedalist*, *Missy Franklin*)} and {(*venue*, *London Aquatic Centre*), (*goldMedalist*, *Emily Seebohm*), (*bronzeMedalist*, *Missy Franklin*)}, respectively. Let $\mathrm{Sim}_{\mathrm{hier}}(e_1, e_2) = 1 - d_{\mathrm{tax}}(e_1, e_2)$. The most similar pair to (*venue*, *London Aquatic Centre*) is itself and with similarity value of 1.0. The most similar pair to (*silverMedalist*, *Emily Seebohm*) is (*goldMedalist*, *Emily Seebohm*) with a similarity value of 0.5. This similarity value is result of the product between $\mathrm{Sim}_{\mathrm{hier}}$(*Emily Seebohm*, *Emily Seebohm*), whose result is 1.0, and $\mathrm{Sim}_{\mathrm{hier}}$(*goldMedalist*, *silverMedalist*), whose result is 0.5. Similarly, the most similar pair to (*goldMedalist*, *Missy Franklin*) is (*bronzeMedalist*, *Missy Franklin*) with a similarity value of 0.5. Thus, the similarity between neighborhoods of *Women's 100 m backstroke* and *Women's 4x100 m freestyle* is computed as $\mathrm{Sim}_{\mathrm{neigh}} = \frac{(1+0.5+0.5)+(1+0.5+0.5)}{3+3} = \frac{4}{6} = 0.667$.

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 information whenever they are used in a corpus similarly. 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 in the knowledge graph. 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 $Incident(e) = \{e_i | (e_i, r, e) \in E\}$. Then, GARUM computes the information shared by two entities using the following formula:

$$\mathbf{Sim_{shared}}(e_1, e_2) = \frac{|\mathrm{Incident}(e_1) \cap \mathrm{Incident}(e_2)|}{|\mathrm{Incident}(e_1) \cup \mathrm{Incident}(e_2)|}, \tag{3}$$

The values depends on how much informative or specific are the compared entities. For example, an entity representing *London Aquatic Centre* is included in several neighborhoods in a knowledge graph like DBpedia. This means that *London Aquatic Centre* is not a specific entity. This is reflected in the denominator of $\mathrm{Sim}_{\mathrm{shared}}$. Thus, abstract or non-specific entities require a greater amount of co-occurrences in order to obtain a high value of similarity. In Fig. 1, entities *Emily Seebohm*, *Missy Franklin*, and *London Aquatic Centre* have incident edges. *London Aquatic Centre* have five incident edges, while *Emily Seebohm* and *Missy Franklin* have four and three, respectively. *Emily Seebohm* and *Missy Franklin* co-occurs in three neighborhoods. Thus, $\mathrm{Sim}_{\mathrm{shared}}$ returns a value of $\frac{3}{4} = 0.75$.

London Aquatic Centre is included in five neighborhoods in sub-graph showed in Fig. 1. However, it is included in the neighborhood of each sport event located in this venue in the full graph of DBpedia.

Attributes: Entities in knowledge graphs are related with other entities and with attributes through datatype properties, e.g., temperature or protein sequence. GARUM 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 [23] for genes or Jaro-Winkler for strings, GARUM does not rely on a specific measure to compare attributes. Depending on the domain, users should choose a similarity measure for each type of attribute. Figure 1 depicts the entity representing *Women's 4x100 m medley relay*; it has attributes *competitors* and *games*, while *Women's 4x100 m freestyle* has only the attribute *competitors*. Thus, Sim_{attr} between these entities only considers the attribute *competitors*.

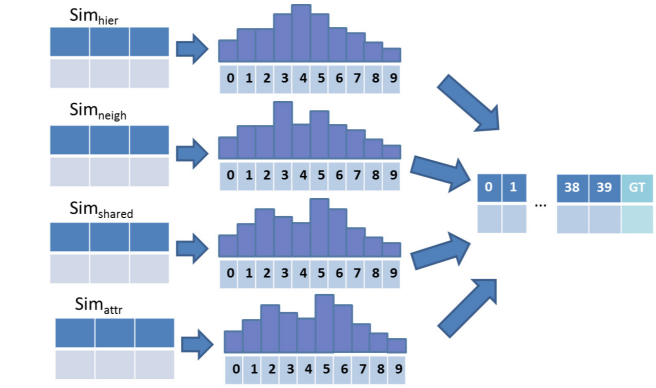
Aggregation Functions: GARUM combines four individual similarity measures and returns a similarity value that aggregates the relatedness among two compared entities. The aggregation function can be manually defined or computed by a supervised machine learning algorithm like a regression algorithm. A regression algorithm receives 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}. The dependent variable is defined by a gold standard similarity measure, e.g., a crowd-funded similarity value. Thus, a regression algorithm produces as output a function $\alpha : X^n \rightarrow Y$, where X^n represents the predictors and Y corresponds to the dependent variable. Hence, GARUM is defined in terms of a function α:

$$\text{GARUM}(e_1, e_2) = \alpha(\text{Sim}_{\text{hier}}, \text{Sim}_{\text{neigh}}, \text{Sim}_{\text{shared}}, \text{Sim}_{\text{attr}}) \tag{4}$$

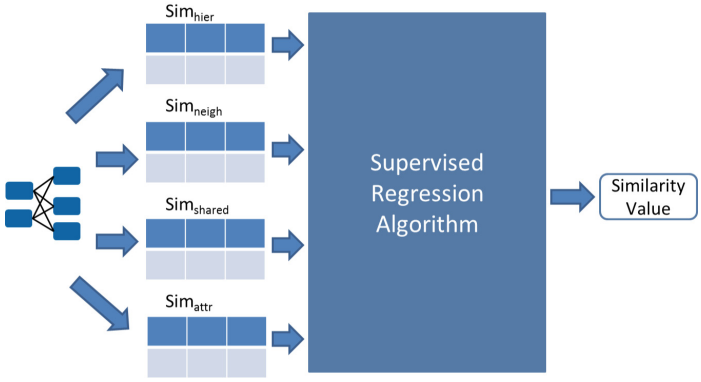
Depending on the regression type, α can be a linear or a non-linear combination of the predictors. In both cases and regardless the used regression algorithm, α is computed by minimizing a loss function. In the case of GARUM, the loss function is the mean squared error (MSE) defined as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2, \tag{5}$$

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 α. Hence, the regression algorithm implemented in GARUM learns from a training dataset how to combine the individual similarity measures by means of a function α, such that the MSE among the results produced by α and the corresponding gold standard (e.g., SeqSim, ECC) is minimized. However, gold standards are usually defined for annotation sets, i.e., sets of knowledge graph entities, instead of for pairs of knowledge graph entities. CESSM [18], and Lee50 [13] datasets are good examples of this phenomenon, where real world entities (proteins or texts) are



(a) Combination function for input matrices. For each matrix a 10-positions vector with the corresponding density value is generated. GT represents the ground truth.



(b) Workflow of the supervised regression algorithm

Fig. 3. Training Phase of the GARUM Similarity Measure. (a) Training workflow using a regression algorithm; (b) Transformation of the input matrices into an aggregated value representing the combination of similarity measures

annotated with terms from ontologies, e.g., the Gene Ontology or the DBpedia ontology. Thus, the regression approach receives as input two sets of knowledge graph entities as showed in Fig. 3(b). Based on these sets, a similarity matrix for each individual similarity measure is computed. The output represents the aggregated similarity value computed by the estimated regression function α. 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 3(a) introduces the matrix transformation. For each matrix, a density histogram with 10 bins is created. Thus, the input dimensions are fixed to $10 \times |\text{Individual similarity measures}|$. In Fig. 3(b), the input consists

of 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 (i.e., GT in Fig. 3(a)).

4 Experimental Results

We empirically evaluate the accuracy of GARUM in three different knowledge graphs. We compare GARUM with state-of-the-art approaches and measure the effectiveness comparing our results with available gold standards. For each knowledge graph, we provide a manually defined aggregation function α, as well as the results obtained using Support Vector Machines as supervised machine learning approach to compute the aggregation function automatically.

Research Questions: We aim at answering the following research questions: **(RQ1)** Does semantics encoded in *entity characteristics* improve the accuracy of similarity values between entities in a knowledge graph? **(RQ2)** Is GARUM able to outperform state-of-the-art similarity measures comparing knowledge graph entities from different domains?

Datasets. GARUM is evaluated on three knowledge graphs: Lee50[4], CESSM-2008[5], and CESSM-2014[6]. Lee50 is a knowledge graph defined by Paul et al. [15] that describes 50 news articles 8 (collected by Lee et al. [13]) with DBpedia entities. Each article has a length among 51 and 126 words, and is described on average with 10 DBpedia entities. The similarity value of each pair of news articles has been rated multiple times by humans. For each pair, we consider the average of human rates as gold standard. CESSM-2008 [18] (see footnote 5) and CESSM-2014 (see footnote 6) consist of proteins described in a knowledge graph with Gene Ontology (GO) entities. CESSM-2008 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 knowledge graph of CESSM-2008 contains 1,908 distinct GO entities and the graph of 2014 includes 3,909 GO entities. The quality of the similarity measures is estimated by means the Pearson's coefficient with respect to three gold standards: SeqSim [23], Pfam [18], and ECC [5] (Table 1).

Implementation. GARUM is implemented in Java 1.8 and Python 2.7; as machine learning approaches, we used the support vector regression (SVR) implemented in the scikit-learn library[7] and a neural network of three layers implemented with the Keras[8] library, both in Python. The experimental study

[4] https://github.com/chrispau1/SemRelDocSearch/blob/master/data/Pincombe_annotated_xLisa.json.
[5] http://xldb.di.fc.ul.pt/tools/cessm/index.php.
[6] http://xldb.fc.ul.pt/biotools/cessm2014/index.html.
[7] http://scikit-learn.org/stable/index.html.
[8] https://keras.io/.

Table 1. Properties of the knowledge graphs used during the evaluation.

Datasets	Comparisons	Ontology
CESSM 2008	13,430	Gene Ontology
CESSM 2014	22,302	Gene Ontology
Lee50	1,225	DBpedia

was executed on an Ubuntu 14.04 64 bits machine with CPU: Intel(R) Core(TM) i5-4300U 1.9 GHz (4 physical cores) and 8 GB RAM. To ensure the quality and correctness of the evaluation, both datasets are split following a 10-cross fold validation strategy. Apart from the machine learning based strategy, since entities (proteins and documents) are described with ontology terms from the Gene ontology or the DBpedia ontology, we manually define two aggregation strategies. Let $A \subseteq V$ and $B \subseteq V$ be set of knowledge graph entities. In the first aggregation strategy, we maximize the similarity value of sim(A, B) using the following formula:

$$\text{sim}(A,B) = \frac{\sum_{i=0}^{|A|} \max_{e_x \in B} \text{GARUM}(e_i, e_x) + \sum_{j=0}^{|B|} \max_{e_x \in A} \text{GARUM}(e_j, e_x)}{|A| + |B|}$$

In the second aggregation strategy, we perform a 1-1 maximum matching implemented with the Hungarian algorithm [11], such that each knowledge graph entity e_i in A is matched with one and only one knowledge graph entity e_j in B; the following formula of sim(A, B) is maximized:

$$\text{sim}(A,B) = \frac{2 \cdot \sum_{(e_i,e_j) \in \text{1-1 Matching}} \text{GARUM}(e_i, e_j)}{|A| + |B|}$$

The first aggregation strategy is used in knowledge graphs Lee50, while the 1-1 matching strategy is used in CESSM-2008 and CESSM-2014.

4.1 Lee50: News Articles Comparison

We compare pairwise the 50 news articles included in Lee50, and consider the knowledge encoded in the hierarchy, the neighbors, and the shared information. Knowledge encoded in attributes is not taken into account. Particularly, we define the aggregation function $\alpha(e_1, e_2)$ as follows:

$$\alpha(e_1,e_2) = \frac{\text{Sim}_{\text{hier}}(e_1,e_2) \cdot \text{Sim}_{\text{shared}}(e_1,e_2) + \text{Sim}_{\text{neigh}}(e_1,e_2)}{2} \tag{6}$$

where $\text{Sim}_{\text{hier}} = 1 - d_{\text{tax}}$.

Results in Table 2 suggest that GARUM outperforms the evaluated similarity measures in terms of correlation. Though d_{ps} obtains alone better results than

d_{tax}, its combination with the other two individual similarity measures delivers worse results. Further, we observe that the aggregation function obtained by the SVR and NN approaches outperforms the manually defined aggregation function.

Table 2. Comparison of Similarity Measures. Pearson's coefficient of similarity measures on the Lee et al. knowledge graph [13]; highest values in **bold**

Similarity measure	Pearson's coefficient
LSA [12]	0.696
SSA [7]	0.684
GED [20]	0.63
ESA [6]	0.656
d_{ps} [16]	0.692
d_{tax} [1]	0.652
$\text{GBSS}_{r=1}$ [15]	0.7
$\text{GBSS}_{r=2}$ [15]	0.714
$\text{GBSS}_{r=3}$ [15]	0.704
GARUM	**0.727**
GARUM SVR	**0.73**
GARUM NN	**0.74**

4.2 CESSM: Protein Comparison

CESSM knowledge graphs are used to compare proteins based on their associated GO annotations. GARUM considers the hierarchy, the neighborhoods, and the shared information as *entity characteristics*. In this knowledge graph, the different characteristics are combined automatically by SVR and with the following manually defined function:

$$\alpha(e_1, e_2) = \text{Sim}_{\text{hier}}(e_1, e_2) \cdot \text{Sim}_{\text{neigh}}(e_1, e_2) \cdot \text{Sim}_{\text{shared}}(e_1, e_2),$$

where $\text{Sim}_{\text{hier}} = 1 - d_{\text{tax}}$.

Table 3 reports on the correlation between state-of-the-art similarity measures and GARUM with the gold standards ECC, Pfam, and SeqSim on CESSM 2008 and 2014. The correlation is measured with the Pearson's coefficient. The top-5 values are highlighted in gray, and the highest correlation with respect to each gold standard is highlighted in bold. We observe that GARUM SVR and GARUM are the most correlated measures with respect to the three gold standard measures in both versions of the knowledge graph, 2008 and 2014. However, GARUM SVR obtains the highest correlation coefficient in CESSM 2008, while GARUM NN has the highest correlation coefficient for SeqSim in 2014[9].

[9] Due to the lack of training data GARUM could not be evaluated in CESSM 2014 with ECC and Pfam.

Table 3. Comparison of Similarity Measures. 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*. The similarity measures are: 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), Jiang & Conrath's Best-Match Average (JB). GARUM SVR and NN could not be executed for ECC and Pfam in CESSM 2014 due to lack of training data.

Similarity measure	2008			2014		
	SeqSim	*ECC*	*Pfam*	*SeqSim*	*ECC*	*Pfam*
GI [17]	0.773	0.398	0.454	0.799	0.458	0.421
UI [17]	0.730	0.402	0.450	0.776	0.470	0.436
RA [19]	0.406	0.302	0.323	0.411	0.308	0.264
RM [21]	0.302	0.307	0.262	0.448	0.436	0.297
RB [3]	0.739	0.444	0.458	0.794	0.513	0.424
LA [14]	0.340	0.304	0.286	0.446	0.325	0.263
LM [21]	0.254	0.313	0.206	0.350	0.460	0.252
LB [3]	0.636	0.435	0.372	0.715	0.511	0.364
JA [8]	0.216	0.193	0.173	0.517	0.268	0.261
JM [21]	0.234	0.251	0.164	0.342	0.390	0.214
JB [3]	0.586	0.370	0.331	0.715	0.451	0.355
d_{tax} [1]	0.650	0.388	0.459	0.682	0.434	0.407
d_{ps} [16]	0.714	0.424	0.502	0.75	0.48	0.45
OnSim [26]	0.733	0.378	0.514	0.774	0.455	0.457
IC-OnSim [25]	0.779	0.443	0.539	0.81	0.513	0.489
GARUM	0.78	0.446	0.539	0.812	**0.515**	**0.49**
GARUM SVR	**0.86**	**0.7**	**0.7**	0.864	-	-
GARUM NN	0.85	0.6	0.696	**0.878**	-	-

5 Related Work

Several similarity measures have been proposed in the literature to determine the relatedness between knowledge graph entities; they exploit knowledge encoded in different *entity characteristics* in the knowledge graph including: hierarchies, length and amount of the paths among entities, or information content.

The measures d_{tax} [1] and d_{ps} [16] only consider hierarchies of a knowledge graph during the comparison of knowledge graph entities. These measures compute similarity values based on the relative distance of entities to their lowest common ancestor. Depending on the knowledge graph, different relation types may represent hierarchical relations. In OWL ontologies *owl:subClassOf* and *rdf:type* are considered the main hierarchical relations. However, in some knowledge graphs such as DBpedia [4], other relations like *dct:subject*, can be also regarded as hierarchical relations. PathSim [24] and HeteSim [22] among others consider only the neighbors during the computation of the similarity between two entities in a knowledge graph. They compute the similarity between two

entities based on the number of existing paths between them. The similarity value is proportional to the number of paths between the compared entities. Unlike GARUM, PathSim and HeteSim do not distinguish between relation types and consider all relation types in the same manner, i.e., knowledge graphs are regarded as pairs $G = (V, E)$, where edges are not labeled. GBSS [15] considers two of the identified entity characteristics: the hierarchy and the neighbors. Unlike PathSim and HeteSim, GBSS distinguishes between hierarchical and *transversal* relations[10]; they also consider the length of the paths during the computation of the similarity. The similarity between two entities is directly proportional to the number of paths between these entities. Shorter paths have higher weight during the computation of the similarity. Unlike GARUM, GBSS does not take into account the property types that relate entities with their neighbors.

Information Content based similarity measures rely on specificity and hierarchical information [8,14,19]. These measures determine relatedness between two entities based on the Information Content of their lowest common ancestor. The Information Content is a measure to represent the generality or specificity of a certain entity in a dataset. The greater the usage frequency, the more general is the entity and lower is the respective Information Content value. Contrary to GARUM, these measures do not consider knowledge encoded in other *entity characteristics* like neighborhood. OnSim and IC-OnSim [25,26] compare ontology-based annotated entities. Though both measures rely on neighborhoods of entities and relation types, they require the execution of an OWL reasoner to obtain inferred axioms and their justifications. These justifications are taken into account for determining relatedness of two annotated entities. Thus, OnSim and IC-OnSim can be costly in terms of computational complexity. The worst case for the classification task with an OWL2 reasoner is 2NEXP-Time [9]. GARUM does not make use of justifications, which reduces significantly the execution time and allows for its use in non-OWL graphs.

6 Conclusions and Future Work

We define GARUM a new semantic similarity measure for entities in knowledge graphs. GARUM relies on knowledge encoded in *entity characteristics* to compute similarity values between entities and is able to determine automatically aggregation functions based on individual similarity measures and a supervised machine learning algorithm. Experimental results suggest that GARUM is able to outperform state-of-the-art similarity measures obtaining more accurate similarity values. Further, observed results show that the machine learning approach is able to find better combination functions than the manually defined functions.

In the future, we will evaluate the impact of GARUM in data-driven tasks like clustering or search and in to enhance knowledge graph quality, e.g., link discovery, knowledge graph integration, and association discovery.

[10] Transversal relations correspond to object properties in the knowledge graph.

Acknowledgements. This work has been partially funded by the EU H2020 Programme for the Project No. 727658 (IASIS).

References

1. Benik, J., Chang, C., Raschid, L., Vidal, M.-E., Palma, G., Thor, A.: Finding cross genome patterns in annotation graphs. In: Bodenreider, O., Rance, B. (eds.) DILS 2012. LNCS, vol. 7348, pp. 21–36. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31040-9_3
2. Gene Ontology Consortium, et al.: Gene ontology consortium: going forward. Nucleic Acids Res. **43**(D1), D1049–D1056 (2015)
3. Couto, F.M., Silva, M.J., Coutinho, P.M.: Measuring semantic similarity between Gene Ontology terms. Data Knowl. Eng. **61**(1), 137–152 (2007)
4. Damljanovic, D., Stankovic, M., Laublet, P.: Linked data-based concept recommendation: comparison of different methods in open innovation scenario. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 24–38. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30284-8_9
5. Devos, D., Valencia, A.: Practical limits of function prediction. Prot.: Struct. Funct. Bioinform. **41**(1), 98–107 (2000)
6. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: IJCAI, vol. 7, pp. 1606–1611 (2007)
7. Hassan, S., Mihalcea, R.: Semantic relatedness using salient semantic analysis. In: AAAI (2011)
8. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint arXiv:cmp-lg/9709008 (1997)
9. Kazakov, Y.: SRIQ and SROIQ are harder than SHOIQ. In: Description Logics. CEUR Workshop Proceedings, vol. 353. CEUR-WS.org (2008)
10. Köhler, S., et al.: The Human Phenotype Ontology project: linking molecular biology and disease through phenotype data. Nucleic Acids Res. **42**(D1), D966–D974 (2014)
11. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Log. Q. **2**(1–2), 83–97 (1955)
12. Landauer, T.K., Laham, D., Rehder, B., Schreiner, M.E.: 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, pp. 412–417 (1997)
13. Lee, M., Pincombe, B., Welsh, M.: An empirical evaluation of models of text document similarity. In: Cognitive Science (2005)
14. Lin, D.: An information-theoretic definition of similarity. In: ICML, vol. 98, pp. 296–304 (1998)
15. Paul, C., Rettinger, A., Mogadala, A., Knoblock, C.A., Szekely, P.: Efficient graph-based document similarity. In: Sack, H., Blomqvist, E., d'Aquin, M., Ghidini, C., Ponzetto, S.P., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9678, pp. 334–349. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-34129-3_21
16. Pekar, V., Staab, S.: Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision. In: Proceedings of the 19th International Conference on Computational Linguistics, vol. 1, pp. 1–7. Association for Computational Linguistics (2002)

17. Pesquita, C., Faria, D., Bastos, H., Falcão, A., Couto, F.: Evaluating go-based semantic similarity measures. In: Proceedings of 10th Annual Bio-Ontologies Meeting, vol. 37, p. 38 (2007)
18. Pesquita, C., Pessoa, D., Faria, D., Couto, F.: CESSM: collaborative evaluation of semantic similarity measures. JB2009: Chall. Bioinform. **157**, 190 (2009)
19. Resnik, P., et al.: Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. J. Artif. Intell. Res. (JAIR) **11**, 95–130 (1999)
20. Schuhmacher, M., Ponzetto, S.P.: Knowledge-based graph document modeling. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 543–552. ACM (2014)
21. Sevilla, J.L., et al.: Correlation between gene expression and GO semantic similarity. IEEE/ACM Trans. Comput. Biol. Bioinform. **2**(4), 330–338 (2005)
22. Shi, C., Kong, X., Huang, Y., Yu, P.S., Wu, B.: HeteSim: a general framework for relevance measure in heterogeneous networks. IEEE Trans. Knowl. Data Eng. **26**(10), 2479–2492 (2014)
23. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. **147**(1), 195–197 (1981)
24. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: PathSim: meta path-based top-k similarity search in heterogeneous information networks. In: VLDB 2011 (2011)
25. Traverso-Ribón, I., Vidal, M.: Exploiting information content and semantics to accurately compute similarity of GO-based annotated entities. In: IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB, pp. 1–8 (2015)
26. Traverso-Ribón, I., Vidal, M.-E., Palma, G.: OnSim: a similarity measure for determining relatedness between ontology terms. In: Ashish, N., Ambite, J.-L. (eds.) DILS 2015. LNCS, vol. 9162, pp. 70–86. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21843-4_6