Semantic Web 0 (0) 1
IOS Press

A Semantic similarity measure for predicates in Linked Data

Rajeev Irny [a], P Sreenivasa Kumar [b] and
[a] *Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai, India*
E-mail: rajeeviv@cse.iitm.ac.in
[b] *Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai, India*
E-mail: psk.iitm.ac.in

Abstract. Semantic similarity measures are used in several applications like link-predication, entity summarization, knowledge-base completion, clustering. In this paper, we propose a new semantic similarity measure called Predicate Semantic Similarity (*PSS*), specifically for predicates in linked data. Accounting for the apparent similarity between a pair of inverse predicates such as influences and influenced – by is one of the motivations for the work. We exploit implicit semantic information present in linked data to compute two quantities that capture context and (semantic) proximity aspects of a given pair of predicates, respectively. We build on the Normalized Semantic Web Distance (*NSWD*) and generalise it to predicates to take care of the context aspect. We also propose a novel measure based on neighbourhood-formation computation on a bipartite graph of predicates and classes to capture the proximity aspect. Thus we compute similarity along two semantic-facets namely context and proximity. A weighted sum of these gives us the new measure *PSS*. Through experiments, we evaluate the performance of *PSS* against the existing similarity measures including RDF2Vec. We find that including only one of context or proximity is insufficient. We create ground-truths to facilitate a thorough evaluation. The results indicate that *PSS* improves over all the existing measures for semantic similarity between predicates.

Keywords: Linked data, Context, Semantic similarity, Predicate Similarity

1. Introduction

A semantic similarity measure for entities is a crucial component in several challenging problems like knowledge base completion and entity linking [1], canonicalization of knowledge bases [2], entity summarization [3], to name a few. Existing predicate similarity measures in linked data [4, 5] are driven by the expectation of finding equivalent (or synonymous) and meronomic (e.g isPartOf) predicates. For instance, a typical measure of predicate similarity would result in a high similarity score for a predicate pairs like dbo:citizenship and dbo:nationality which are candidates for equivalence, but would have a low similarity score for predicate pairs like dbo:influenced and dbo:influencedBy. We know that dbo:influenced and dbo:influencedBy are inverses of each other (as stated in DBpedia Ontology[1]). As such, inverse pairs of predicates in linked data can be considered semantically similar. We consider such predicate pairs to be similar as well, this is because a pair of inverse predicates p_i, p_j express similar semantics as the triples $\langle s, p_i, o\rangle$ and $\langle o, p_j, s\rangle$ convey the same semantic information.

The importance of predicate similarity measures can be appreciated by their use in several applications like finding equivalent predicates [5], fusing knowledge cards across search engines [6] and to identify similar concepts in an ontology based on similar predicates [7]. To this end,

[1] http://downloads.dbpedia.org/2016-04/dbpedia_2016-04.nt

we propose a semantic similarity measure exclusively for predicates in linked data called Predicate Semantic Similarity (*PSS*).

Our Contributions

Our contributions in this paper involve computing similarity between predicates along two facets namely *context* and *proximity*. We propose *context* and *proximity* based similarity between predicates in linked data as follows:

- **Context-based Similarity:** We introduce the notion of *contexts* for each predicate as the set of distinct class-types of instances that occur in the subject/object position of triples containing the predicate and *shared-contexts* for a pair of predicates in linked data. It is based on the premise that: similar predicates have a large amount of co-occurrence of *shared-contexts*. To compute the context similarity we adapt the Normalized Semantic Web Distance (*NSWD*) based similarity measure[8]. Note that *NSWD* computes similarity between entities and not between predicates.
- **Proximity-based similarity:** We introduce the notion of proximity based similarity which involves computing pair-wise proximity scores for predicates in linked data. Having computed the proximity scores for each pair of predicate, we assert that similar predicates have similar distribution of proximity scores. To this end, we represent each predicate as a vector such that the j^{th} component of the vector representation for predicate p_i represents its proximity to predicate p_j. The proximity scores are computed by Neighbourhood Formation (*NF*) operation[9] on a bipartite graph with classes and predicates on either side.

  We find that these measures used individually are not effective in capturing the similarity of predicates. The proposed predicate similarity measure (*PSS*) is a weighted sum of the above measures.

The rest of the paper is structured as follows, in Section 2 we discuss current similarity measures for predicates in linked data. In Section 3 we discuss Normalized Semantic Web Distance (*NSWD*) and Neighbourhood Formation (*NF*) as they are necessary to understand this work. In Section 4 we formalize the meaning of *contexts* and explain how semantic similarity can be computed along different dimensions. In Section 5 we show the evaluate the effectiveness of *PSS* through experiments. We end the paper by discussing the conclusion and stating future extensions in section 6.

2. Related Work

Predicate similarity measures are a useful in several applications that use linked data. Wang et al. [6] use predicate similarity to align predicates across knowledge cards and model it as a ontology alignment task. The predicates are usually checked for lexical and semantic similarity. The WordNet based measures like WUP[10] are used to determine semantic similarity between entities and predicates alike. Such a measure depends on the presence of the predicate in the taxonomy, but since linked data often follows arbitrary naming schemes, WordNet taxonomy based measures can prove to be unreliable. For the same reason, similarity measures that check for lexical similarity are also unreliable. Also, such measures have no provision to consider context of occurrence of a predicate in the linked data.

Zhang et al. [5] introduce a unsupervised method to determine local-clusters of equivalent predicates specific to a concept or a class. As a result, predicate pairs that are equivalent w.r.t one class, may not be equivalent w.r.t some other class.

Fu et al. [4] present a semantic similarity measure based on overlap of instances in subject position and overlap of instances object position between the two predicates. In general this measure fails to consistently identify similar predicates under different contexts. We compare our experimental results against this measure. Semantic similarity measures have also been developed for finding semantically similar entities in text. Harispe el al.[11] survey and compare the various state-of-the-art semantic similarity and relatedness measures for predicates in natural language in detail.

RDF2Vec [12] adapts the Word2Vec [13] to represent the entities and predicates in linked data as context-based feature vectors. Ristoski et al.[12] propose performing random walks of fixed length over the RDF graph to obtain graph sub-structures resulting in a sequence of entities and predicates.

These sequences are analogous to word sequences used to train Word2Vec on natural language text. Thus, RDF2Vec chooses to consider the entities and predicates in RDF graph as labels to compute the vector representations of the corresponding entities and predicates in linked data.

Normalized Relevance Distance[14] (NRD) are an adaptation of Normalized Google Distance (NGD)[15]. NGD based distance-measures generally compute co-occurrence between objects (these objects are *terms* in case of NRD, *entities* in case of NSWD, *predicates* in case of SWPD). However, at the heart of such measures lies a frequency function to compute co-occurrence of objects. NRD uses tf-idf scores as the frequency function to compute term-relatedness over documents and interprets this tf-idf measure based co-occurrence score as *relevance*. It is worth noting that in semantic web, we work with *things* not *strings*. As such, NSWD (similar to NRD) makes no effort to exploit representation of entities as *things*. To account for this linked data setting and in order to leverage the class information available in linked data, we introduce Proximity-based similarity (*PS*) (as discussed in Section 4.2). However, since we work with linked data and not textual data, we do not compare NRD with *PSS*.

3. Preliminaries

In linked data, information is modeled as triples of the form $\langle s, p, o\rangle$ where s is the subject, o the object and p the predicate. Excluding literals (like strings, numeric data) in the object position, each entity in the subject and object of a triple can be an instance of one or several *class types*. The set of all triples in a KB can also be visualized as a graph where the entities in subject and object of a triple are the nodes and the predicates are the directed edges from the subject to the object.

In linguistics, the context of a word is the words surrounding it and this context information is used for disambiguating the sense of words. However in linked data the disambiguation is comparatively easier since each entity is represented by a machine-interpretable resource (URI). Moreover, the context information is also be used as a feature to identify semantically similar entities and predicates in linked data.

3.1. Normalized Semantic Web Distance (NSWD)

De Nies et al. [8] define the context of an entity as the set of *entities* that share a predicate with it. Based on this definition of context, they propose a distance measure called Normalized Semantic Web Distance (*NSWD*) between two entities in the KB. *NSWD* is an adaptation of the Normalized Web Distance (*NWD*) [16] for the linked data setting. *NWD* is based on the intuition that if two entities occur together (in web documents) more often than they occur separately, they must be similar. *NSWD* is also based on a similar principle i.e, the more two instances share incoming and outgoing edges as predicates, the more they are similar. *NSWD* is computed as shown in equation 2 where $\lambda \in \{in, out, all\}$, I is the set of instances in the KB and $N = |I|$ i.e the count of all instances in the KB.

$$\begin{aligned} V_{in}(x) &= \{v \in I \,|\, \langle v, p, x\rangle \in \mathrm{KB}\} \\ V_{out}(x) &= \{v \in I \,|\, \langle x, p, v\rangle \in \mathrm{KB}\} \\ V_{all}(x) &= V_{in}(x) \cup V_{out} \end{aligned} \tag{1}$$

$$\begin{aligned} &f_\lambda(x) = |V_\lambda(x)| \textit{ and } f_\lambda(x,y) = |V_\lambda(x) \cap V_\lambda(y)| \\ &NSWD_\lambda(x,y) = \frac{max\{logf_\lambda(x), logf_\lambda(y)\} - logf_\lambda(x,y)}{logN - min\{logf_\lambda(x), logf_\lambda(x)\}} \end{aligned} \tag{2}$$

In other words, $NSWD_\lambda(x,y)$ represents the conditional probability of co-occurrence of instances x and y in the KB. *NSWD* is a distance measure and $NSWD \in [0, \infty)$. De Nies et al normalize it to obtain Sim_{NSWD} [8], a similarity metric such that $Sim_{NSWD} \in [0, 1]$ so that a pair similar predicates p_i, p_j have a higher $Sim_{NSWD}(p_i, p_j)$ score than the dissimilar predicates p_i, p_k.

Note that *NSWD* determines similarity only for pairs of instances and not pairs of predicates in linked data. Since predicates are indispensable to accurately representing knowledge, a semantic similarity measure for predicates in linked data is of significant utility. We build upon the definition of *NSWD* to present a semantic similarity measure for predicates in linked data in Section 4.1.

3.2. Neighbourhood Formation (NF)

Given a bipartite graph $G = \langle V_1 \cup V_2, E\rangle$ and a node $p_i \in V_1$, the Neighbourhood Formation (*NF*) operation[9] involves computing the *proximity scores* of all nodes $p_j \in V_1$ *w.r.t* p_i. E is the set of edges in G from nodes in V_1 to V_2. *NF* operation involves computing neighbourhoods within V_1(or V_2) such that the nodes within a neighbourhood have high proximity scores. The *proximity scores* are computed by performing random-walks with restarts over the graph G. These walks begin at node p_i and during each walk we maintain the frequency of visiting a node $p_j \in V_1$ from p_j. The intuition is that the frequency of visiting a node p_j is proportional to its proximity *w.r.t* p_i. Thus, the *proximity score* for p_j would simply be the probability of visiting p_j from p_i. Subsequently, neighbourhoods of nodes in V_1 can be formed based on their pair-wise *proximity scores*.

The *NF* operation involves modelling the bipartite graph as a $k \times n$ matrix M where $k = |\{V_1\}|$, $n = |\{V_2\}|$ such $M(i, j)$ represents the weight of the edge from a node $p_i \in V_1$ to a node $c_j \in V_2$. Subsequently, an adjacency matrix M_A is created using M as shown in equation (3) and M_A is transformed to a column normalized matrix N_A such that each column sums upto 1.

$$M_A = \begin{pmatrix} 0_{n,k} & M \\ M^T & 0_{k,n} \end{pmatrix} \tag{3}$$

With this setup, we calculate the pair-wise proximity score as shown in Algorithm 1. Here, we represent any node $a \in V_1$ as a $(k+n) \times 1$ dimensional steady-state probability vector $\vec{p_i}$. Initially $\vec{p_i} = \vec{q_i}$. On iterative application of the transformation in line 4, we achieve the steady-state probability vector $\vec{p_i}$. In the algorithm, r is the restart probability for the random walks. Thus, at the steady-state, $\vec{p_i}(i : k)$ in line 6 represents the first k components of $\vec{p_i}$ which contains the proximity scores of all nodes $p_j \in V_1$ such that the value $\vec{p_i}(j)$ is the *proximity score* of p_j *w.r.t* p_i. Sun et. al[9] propose more efficient and scalable variants of the Neighbourhood Formation algorithm which we use in experiments.

Algorithm 1: Neighbourhood formation

Data: node $\vec{p_i}$, Bipartite graph $M_{(k,n)}$, restart probability r, tolerant threshold ε
Result: Vector representation of node $\vec{p_i}$

1 initialize $\vec{q_i}$ as a one-hot vector with $\vec{q_i}(i) = 1$;
2 Construct M_A and N_A matrices.
3 **while** $|\Delta\vec{p_i}| \geqslant \varepsilon$ **do**
4 $\quad \vec{p_i} = (1-r)N_A\vec{p_i} + r\vec{q_i}$
5 **end**
6 return $\vec{p_i}(1 : k)$

4. Semantic Similarity

We propose Predicate Semantic Similarity (*PSS*), a semantic similarity measure for predicates in linked data. It has two facets, one to compute the context-based similarity and other to computes the proximity-based similarity between two predicates. We formally define *context* and describe the context-based similarity measure in Section 4.1. In Section 4.2 we describe the proximity based similarity measure. Together they harness the semantic features of the linked data such as the rdf:type of entities, the neighbourhood of predicates and implicit relationship between classes.

4.1. Context-based similarity

We define the *context* of a predicate as the set of class types of instances in its subject, object. For a predicate p, the sets $C_s(p)$, $C_o(p)$ in equations (5) represent the subject-side and object-side *contexts* respectively and $C_u(p)$ is the union of the two context sets. Given a set of entities S, $Types(S)$ in equation (4) gives the set of distinct class types of the entities in the set S and KB is the set of all triples in linked dataset under consideration and $\langle x$ rdf:type $t\rangle$ indicates that the entity x is an instance of class type t.

$$Types(S) = \bigcup_{x \in S} \{t \mid \langle x \text{ rdf:type } t\rangle \in \text{KB}\} \tag{4}$$

$$
\begin{aligned}
C_s(p) &= Types(\{x \mid \langle x,p,o\rangle \in \text{KB}\}) \\
C_o(p) &= Types(\{x \mid \langle s,p,x\rangle \in \text{KB}\}) \\
C_u(p) &= C_s(p) \cup C_o(p)
\end{aligned} \tag{5}
$$

Similarly, given two predicates p, q their *shared contexts* represents the co-occurrence of the contexts of p and q such that they share common entities in the subject and object of the predicates as shown in equation (6). C_f and C_r represent the two ways in which p and q share contexts. We call C_f the *forward* shared-context since the subject of p is the subject of q while, C_r is called *reverse* shared-context since the subject of p is the object of q and vice-versa.

$$
\begin{aligned}
C_f(p,q) = {} & Types(\{s \mid \langle s,p,o\rangle \in \text{KB}, \langle s,q,o\rangle \in \text{KB}\}) \\
& \cup Types(\{o \mid \langle s,p,o\rangle \in \text{KB}, \langle s,q,o\rangle \in \text{KB}\}) \\
C_r(p,q) = {} & Types(\{s \mid \langle s,p,o\rangle \in \text{KB}, \langle o,q,s\rangle \in \text{KB}\}) \\
& \cup Types(\{o \mid \langle s,p,o\rangle \in \text{KB}, \langle o,q,s\rangle \in \text{KB}\})
\end{aligned} \tag{6}
$$

We use the *context* and *shared context* information and propose an instance based similarity measure called Semantic Web Predicate Distance (*SWPD*) as shown below. A very basic variant of the context-based similarity measure was first proposed in [17].

4.1.1. Semantic Web Predicate Distance (SWPD)

The Semantic Web Predicate Distance is based on the intuition that similar predicates are used in similar *contexts*. We model this intuition as a distance measure which is inspired by the Normalized Semantic Web Distance (*NSWD*). As the name suggests, *SWPD* measures the semantic distance between two predicates, as shown in equations (7) and (8). Here, T is the set of all class types in the linked data. The $SWPD_f$ measures the semantic distance between p, q in the forward direction since it uses the forward shared-context to determine similarity. $SWPD_f$ expresses synonymous, hierarchical relationship. $SWPD_r$ measures the semantic distance in the reverse direction as it uses the reverse shared-context. This helps to account for the inverse relationship between the predicates. In general, we may interpret the *SWPD* as a measure of the co-occurrence of the contexts of two predicates where $SWPD_f$ measures the conventional distance while $SWPD_r$ measures the inverse distance between the two predicates.

$$
SWPD_f(p,q) = \frac{max\{log|C_u(p)|, log|C_u(q)|\} - log|C_f(p,q)|}{log|T| - min\{log|C_u(p)|, log|C_u(q)|\}} \tag{7}
$$

$$
SWPD_r(p,q) = \frac{max\{log|C_u(p)|, log|C_u(q)|\} - log|C_r(p,q)|}{log|T| - min\{log|C_u(p)|, log|C_u(q)|\}} \tag{8}
$$

Example 1. Given a graph G with $|V| = 100$, for the subgraph of G shown in Figure 1, we calculate the $SWPD_f(p, q)$ and $SWPD_r(p, q)$. Here, x_i ($\forall i = 1,..5$) are nodes in the graph and p, q are edges, $:a$ is the $rdf{:}type$ predicate while C_i ($\forall i = 1, 2, 3$) are classes/concepts in the linked data. From equations (4), (5) and (6) we get the following:

$$
\begin{aligned}
C_s(p): &\quad Types(\{x_1, x_2, x_4\}) \\
C_o(p): &\quad Types(\{x_1, x_2, x_3, x_5\}) \\
C_u(p): &\quad \{C_1, C_2, C_3\}
\end{aligned}
$$

i.e we get $C_s(p) = \{C_1, C_2\}$, $C_o(p) = \{C_1, C_2, C_3\}$. Similarly we get $C_u(q) = \{C_1, C_2\}$.

From equation (6) we get:

$$
\begin{aligned}
C_f(p,q): &\quad \{C_1, C_2\} \\
C_r(p,q): &\quad \{C_2\}
\end{aligned}
$$

Here, we obtain C_f since it is the union of $Types(\{x_1, x_2\})$ and $Types(\{x_3, x_4\})$ similarly, we obtain C_r because it is the union of $Types(\{x_2\})$ and $Types(\{x_4\})$.

Based on the *context* and *shared context* sets computed above, from equations (7), (8) we get $SWPD_f = 0.1036$ and $SWPD_r = 0.2808$.

SWPD is a distance measures, thus we expect semantically similar predicates like dbo:influences, dbo:influencedBy or dbo:nationality, dbo:citizenship to be semantically closer to each other and as a result the corresponding $SWPD_r$ or $SWPD_f$ for semantically similar predicates will be small and vice-versa. However, a similarity measure usually provides higher scores for similar predicates so we recalibrate the *SWPD* so that semantically similar predicates have higher score compared to dissimilar predicates.

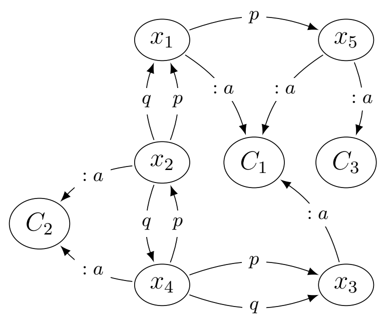

Fig. 1. Example to illustrate *SWPD* scores for predicates p, q

Recalibrating SWPD

Theoretically, we may have $SWPD \geqslant 1$ when C_f (or $C_r = 0$) and $C_u(p) = C_u(q) = T$ (from equation (7)(8)) as a result we have $SWPD \in [0, SWPD_{max}]$. Since similarity scores are generally in $[0,1]$ with higher scores for more similar predicates (and vice-versa), we recalibrate so that the resulting measure is also in the $[0,1]$ range. The details of this recalibration is as discussed in [8]. We call the result of this recalibration as Context Similarity (CS) as shown in equation (9):

$$CS_\lambda(p,q) = \begin{cases} 1 - SWPD_\lambda \times (1 - \frac{1}{SWPD_{max}}) & if\, SWPD_\lambda \in [0,1] \\ (1 - \frac{SWPD_\lambda}{SWPD_{max}}) \times \frac{1}{SWPD_{max}} & if\, SWPD_\lambda \in (1, SWPD_{max}] \end{cases} \tag{9}$$

Here, $\lambda = \{f, r\}$ which means the same recalibration applies to both $SWPD_f$ and $SWPD_r$. Also $SWPD_{max} = log_2(\frac{|T|}{2} + 1)$, an upper-bound on $SWPD_\lambda$ (as shown in [8]). Post the recalibration shown in equation (9) we get $CS(p,q) \in [0,1]$. We get $CS_\lambda(p,q) = 0$ when $SWPD_\lambda(p,q) = 1$, implying that (p,q) are dissimilar as they do not share any context. Similarly we have $CS_\lambda = 1$ when $SWPD_\lambda(p,q) = 0$ which means that (p,q) have a perfect overlap of contexts. Thus, for predicates p, q we consider the maximum of CS_f and CS_r similarity scores as the Context-based similarity score where a higher CS_r value implies the p, q are inverse similar.

4.1.2. Discussion

Distance measures that are based on information content, such as *SWPD*, *NSWD* and *NWD* require a function to calculate the co-occurrence of terms (like predicates, entities or web-pages) and thus compute the semantic distance between terms. However these measures model the co-occurrence simply as intersection of entities in case of [8] terms in case of [16]) and classes in case of *SWPD* as shown in Equation (6)). In doing so, *SWPD* ignores the effect that implicit class axioms (like subsumption, equivalence etc) can have on predicate similarity. For instance, consider the Example 1. In this case, if $C_1 \equiv C_2$ then we have $C_r(p,q) \equiv C_f(p,q)$. Similarly, if $C_2 \sqsubseteq C_3$ then we could have $C_u(p) \equiv C_u(q)$. Under these changes to the context-sets, it would be reasonable to assume that the corresponding $SWPD_f$ (or $SWPD_r$) score will also be affected. This demonstrates that implicit intra-class relationships (i.e class axioms) influence the predicate-similarity. Thus, we need to be aware of the implicit intra-class relationships in a linked dataset while computing predicate similarity.

Consequently, we propose a method to consider the implicit relationship among the classes and predicates in Section 4.2. This method adapts the Neighbourhood Formation approach [9].

4.2. Proximity-based Similarity (PS)

In the previous section we acknowledged that context alone is not enough to compute similarity between predicates, we need to be semantically aware and account for implicit intra-class relationships between classes as well. We attempt to account for these semantic artifacts by measuring the semantic *proximity* between classes. We hypothesize that two classes are in close semantic proximity then they are likely to be implicitly-related. Thus, we interpret the semantic proximity between classes in the context-sets of predicates as a proxy for the implicit intra-class relations.

To compute *proximity* between classes, we condense the linked dataset into two bipartite graphs. Let $\mathcal{P}$ be the set of predicates, $\mathcal{E}$ the set of entities and $\mathcal{C}$ the set of classes in a linked dataset. $\mathcal{E}$, $\mathcal{C}$ and $\mathcal{P}$ are mutually disjoint because an entity cannot be a class or a predicate and vice-versa. We use this fact to represent the relationship between predicates and their *contexts* as a bipartite graph. The following definitions are needed to precisely set-up the framework:

Definition 1. *Consider a bipartite graph $G_s = (\mathcal{P} \cup \mathcal{C}, \mathcal{W}_s)$ where $\mathcal{P} = \{p_i \mid 1 \leqslant i \leqslant k\}$, $\mathcal{C} = \{c_j \mid 1 \leqslant j \leqslant n\}$ are the set of predicates and classes in linked data and form the vertices of G_s. $\mathcal{W}_s$ in G_s is the set of weighted edges such that an edge from $p_i \in \mathcal{P}$ to $c_j \in \mathcal{C}$ means that the class $c_j \in C_s(p_i)$. We call G_s as the "source-side bipartite graph" because the edges in this bipartite graph are determined based on the subject-side context of the predicate p.*

The bipartite graphs G_s is stored as a $k \times n$ matrix M_s, such that $M_s(i,j)$ is the weight of the edge $p_i \leftrightarrow c_j$. Similarly, we construct the bipartite graph G_o, the *object-side bipartite graph* which utilizes the object-side contexts to construct the edges in the bipartite graph. Thus, in this way we have condensed the entire linked dataset into two bipartite graphs G_s and G_o.

The edge weight is the product of *class frequency*(*cf*) and *inverse-class frequency*(*icf*). *cf*-*icf* is similar to *tf-idf* in linguistics, we define *cf* and *icf* as follows:

Definition 2. ***(class frequency)*** *For a given edge $p_i \leftrightarrow c_j$ in G_s, the class frequency* (cf) *is the count of triples where p_i is the predicate and entity in the subject is an instance of class c_j.*

Definition 3. ***(inverse class frequency)*** *For a given edge $p_i \leftrightarrow c_j$ in G_s, the inverse-class frequency for a class c_j is the logarithmically scaled inverse fraction of the number of predicates(p_i) that have c_j in its context $C_s(p_i)$. We interpret* icf *as class-specificity i.e a measure of how often a class appears in the context of a predicate.*

We similarly assign weights to edges in G_o.

Proximity

Now that we have condensed a linked dataset into bipartite graphs, we can compute the proximity scores for predicate pairs. Consider the following operations on G_s.

1. Start at predicate $p_i \in \mathcal{P}$ in G_s, perform random-walks with restarts.
2. Note the frequency of visiting each node p_j from p_i.

Perform the same operation for G_o as well. Based on this operation, we can now define *proximity* as follows:

Definition 4. ***(Proximity):*** *For predicates p, q, proximity of p w.r.t to q is proportional to the probability of visiting q from p, while performing random walks on G_s and G_o.*

Note that for predicates p, q, if the classes in their context-sets are implicitly related to each other (i.e they could be equivalent or could be in related hierarchically) then it is likely that the probability of visiting q from p would be high i.e the intra-class relationships of the classes in the context-sets of predicates p, q translates into a higher probability of visiting q from p and vise-versa. Thus, in-turn we can interpret that q has a higher proximity w.r.t p.

We use the Neighbourhood Formation operation (cf. Section 3.2) to extract the proximity between predicates. Analogous to M_A and N_A in Section 3.2, we create adjacency matrix M_{As}, N_{As} using M_s, and M_{Ao}, N_{Ao} using M_o. We can now apply the Neighbourhood Formation Algorithm (Algorithm 1)) on both G_s and G_o to obtain the

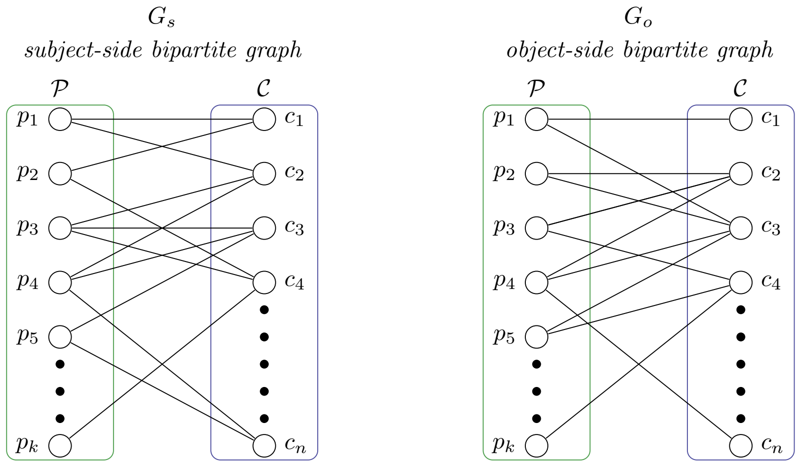

Fig. 2. Bipartite Graphs for the subject-side and object side for predicates in linked data. Neighbourhood Formation operation is performed on both the bipartite graphs to obtain proximity-scores.

pair-wise subject-side proximity and object-side proximity-scores for all the predicates respectively. The bipartite graph G_s is shown in Figure 2.

For a predicates $p, q \in \mathcal{P}$, let the subject-side steady state probability vectors computed from Algorithm 1 be $\vec{p^s}$ and $\vec{q^s}$ respectively. Similarly $\vec{p^o}$ and $\vec{q^o}$ are the object-side steady-state probability vectors for predicates p, q. Based on these vector representations for predicates p, q we obtain the proximity-based similarity as geometric-mean of the cosine-similarity between the subject-side and object-side vectors as shown in Equation 10

$$
\begin{aligned}
RS(p,q) &= \sqrt{sim(\vec{p^s}, \vec{q^s}) * sim(\vec{p^o}, \vec{q^o})} \\
where,\ sim(\vec{p}, \vec{q}) &= \frac{\vec{p} \bullet \vec{q}}{\|\vec{p}\|\,\|\vec{q}\|}\ (cosine\text{-}similarity)
\end{aligned}
\tag{10}
$$

Discussion The random walks with restarts over the G_s and G_o ensure that the proximity-scores are influenced by the implicit relationships between the classes in the context of a predicate. This is because the (implicit) relationship that exists between classes like equivalence, subsumption manifests as in-coming edges from the classes $\mathcal{C}$ to predicates $\mathcal{P}$. For instance, consider the sub-graph in Example 1. Let us assume that $C_1 \sqsubseteq C_3$, then under this setting, every predicate $p_i \in G_s$ (or G_o) with edge to C_1 will most likely have an edge to C_3 as well and any predicate $p_j \in G_s(\mathcal{P})$ (or $G_o(\mathcal{P})$) will also have corresponding edges to C_1, C_3 provided p_i, p_j are similar. We observe that implicit relationships among classes like the one mentioned above essentially assist in highlighting the similarity between predicates in linked data.

4.3. *Predicate Semantic Similarity (*PSS*)*

The Context-based similarity (CS) and proximity-based similarity (PS) complement each other because CS measures the similarity between predicates based on the information content of their *contexts* and *shared-contexts* while PS measures the similarity based on the relationships that hold between the classes in the *context* of predicates. Thus, having computed CS and PS, the semantic similarity between any two predicates is the weighted sum of PSS and PS as shown in equation (11). For experiments we use $\alpha = 0.5$, giving equal importance to both CS and PS.

$$PSS(p,q) = \alpha CS(p,q) + (1-\alpha) PS(p,q) \tag{11}$$

Thus, the average of CS and PS similarity scores is called Predicate Semantic Similarity (PSS). We compare PSS with other similarity measures in Section5.

5. Evaluation

In this section we describe the datasets, ground-truth, evaluation protocol. We also compare th performance of our work against a baseline and several existing similarity measures. Finally, we discuss the evaluation results in Section5.3 where we also discuss the contribution of *CS* and *PS* to *PSS*.

5.1. Dataset and Ground Truth

Dataset Used For experiments we use the DBpedia 2016-04 infobox properties, GeoSpecies and Semantic Web DogFood (SWDF) linked datasets. We have pre-processed these datasets so that they contain only object-properties. The DBpedia dataset contains entities belonging to the dbo[2] namespace. The GeoSpecies and SWDF dataset however contains entities from several namespaces like foaf, swc, swrc, rdfs etc. The SWDF dataset contains facts about several conferences and workshops. Table1 contains dataset specific information. Thus, from the sizes of the datasets, it is clear that *PSS* as a similarity measure can be applied to large, medium and small sized datasets and is thus scalable.

Constructing the Ground Truth Due to lack of publicly available resources, we manually constructed the ground-truth for each dataset. Ideally, a ground-truth should contain a diverse sample of predicates pairs such that some pairs are very similar while some are dissimilar. We construct the ground-truth based on this principle. We begin by clustering the set of predicates in each dataset. The distance measure for clustering is the count of common $\langle subject, object \rangle$ pairs shared between predicates. This means two predicates will belong to a cluster if the count of $\langle subject, object \rangle$ pairs they share is above a certain threshold. Thus, applying clustering to the set of predicates gave us several clusters of predicates. Now, to construct the ground-truth, we select predicate pairs from within as well as across the clusters. This ensures that predicate pairs that belong to the same cluster are more likely to be similar since they share a greater number of entities while predicate pairs that belong to different clusters are less likely to be similar. Accordingly, the ground-truth contains dissimilar predicate pairs as well.

Finally, having selected the predicate pairs, we now need to assign similarity scores to each of them. The task of assigning similarity scores was performed by a group of 3 human-evaluators. These evaluators were required to assign a similarity score on the scale of $1-5$ for each predicate pair in the ground-truth. The final similarity score for the predicate pairs in a ground-truth is averaged over all evaluators. The ground-truth for the datasets, human evaluations and other resources are available online.[3]

5.2. Evaluation Protocol

In this section, we evaluate the accuracy and the quality of the results generate by *PSS*. We compare the accuracy of *PSS* against the existing similarity measure such as WUP[18] (a WordNet based similarity measure), Data-driven similarity measures such as Jaccard Similarity which measures the overlap of $\langle subject, object \rangle$ between predicates and Fu et al[4]. We also compare compare against RDF2Vec [12]. Since RDF2Vec provides latent representations of predicates in vector form, the similarity between two predicates can be computed easily. To measure accuracy, we take a random sample of predicate pairs from the ground-truth for each dataset. The number of predicate pairs in the sample for evaluation for DBpedia, GeoSpecies and SWDF are 33, 10 and 10 respectively. For each predicate pair in the sample, each of the similarity measures provide a similarity score. We quantify the performance of each similarity measure by computing the Pearson's Correlation Coefficient w.r.t the ground-truth. The correlation coefficient ranges from -1 to 1 where a positive value implies a positive correlation and vice-versa. Thus, higher the correlation scores better the performance of the similarity measure under consideration.

We evaluate performance of *PSS* qualitatively as well. This enables us to examine the quality of the results generated by *PSS*. We do this by taking a random sample of 5 predicates from DBpedia. For each predicate in the sample, we generate the top-k most-similar predicates based on the *PSS* scores. For each predicate in the random sample, a group of 3 experts each generate a ranked-list of top-k most similar predicates. These ranked-list of predicates form the basis of evaluating the quality of *PSS* as a similarity measure. To quantify the performance of this task, we use the Spear-

[3] https://bit.ly/2uPISpt

Table 1
Details of datasets used in evaluation

Dataset	#Properties	#Classes	#Entities	#Triples
DBpedia	652	461	5461016	58437974
GeoSpecies	97	41	104756	1631504
SWDF	86	79	13522	89159

man's Footrule metric[19]. This metric provides the distance between two ranked-lists which contain the same set of items. It is formally defined in equation (12) where $\sigma_j(i)$ represents the rank of item i in list j. F_{max} is $\frac{n^2}{2}$ (when n is even) and $\frac{(n+1)(n-1)}{2}$ (when n is odd).

$$F(\sigma_1, \sigma_2) = \Sigma_{i=1}^{n} |\sigma_1(i) - \sigma_2(i)|$$
$$F_N = 1 - \frac{F}{F_{max}} \; (Normalization) \tag{12}$$

F essentially measures the distance between two ranked-lists. We get $F = 0$ for identical lists. The F distance is normalized (equation (12)) so that identical lists have $F_N = 1$, doing so facilitates simpler evaluation. Thus, we can obtain F_N scores for each of the ranked-lists of top-k similar predicates generated by experts and compare it against the top-k list produced using *PSS*.

Throughout the experiments, we set: the restart probability r as 0.15, the tolerant-threshold for random-walks ε as 0.01 and contribution of *CS* in *PSS* α as 0.5.

5.3. Evaluation Results

Table 2 compares the Pearson correlation coefficient of several similarity measures w.r.t the ground-truth. It is evident from the results that context-based measures like *PSS* and RDF2Vec perform the better than Data-driven and WordNet based measures. *PSS* performs better than WordNet based measures since these measures are primarily fine-tuned for computing similarity between entities and not predicates in linked data. Also. such measures employ naive techniques to distinguish between entities and predicates (using sense/parts-of-speech of a word) and thus even though WordNet based measures utilize semantics encoded data, they fail to perform well in linked data-setting.

Like the WordNet based measures, data-driven measure compute similarity with the expectation of determining equivalent (i.e synonymous) or meronimic (is-part-of) predicates. Thus, because of this expectations, they fail to identify similarity between inverse predicates like ⟨dbo:previousWork, dbo:nextWork⟩.

We compare *PSS* against both the models of RDF2Vec i.e Skip-gram (SG) and Continuous Bag-of-words (CBOW). Ristoski et al [12] in RDF2Vec capture the context of entity/predicates by performing random walks of fixed lengths over RDF graphs. Thus, this supports our claim that context is critical in computing similarity between predicates. Also, from Table2, Context-based similarity (*CS*) performs as good as RDF2Vec. While both *CS* and RDF2Vec capture the context of predicates, *CS* being exclusively for predicates is able to model the context-information better and thus produces better results. Also, the embeddings of entities/predicates in RDF2Vec do not leverage the semantic information in the form of implicit relationships that may exist among the classes and predicates. From the results in Table2 it is evident that when we augment a context-based similarity measure (*CS*) with such information (i.e complement it with proximity-based similarity *PS*) the resulting *PSS* has better accuracy.

The entries for RDF2Vec corresponding to GeoSpecies and SWDF are empty since we could not generate the corresponding latent-representations of entities and predicates in these datasets. Similarly owing to the highly specialized domain of GeoSpecies and SWDF, the WordNet based WUP measures could not generate usable results, hence the corresponding entries in Table2 are left blank.

***Impact of* PS** Correlation scores in Table2 show that the when *PS* is combined with *CS* in *PSS*, the resulting performance improves. This happens despite the fact the *PS* has negative correlation w.r.t the ground-truth in some cases. Such an outcome is as expected because *PS* introduces new information in the form of implicit relationships between the classes in the *context* of a predicate, and we've seen that complementing *CS* with such information leads to overall improvement in performance. Thus, *PS* has a positive impact on computing similarity between predicates.

Comparison with Baselines

We also compare *PSS* against two baselines. The objective of this comparison is to emphasize the capability of *CS* (context-based similarity) in measuring similarity between predicates.

- In Baseline#1, to compute the pairwise similarity between predicates, we represent each predicate as $(1{\times}N)$ dimensional vector where N is the count of classes in a dataset. For a predicate p_j, the i^{th} component of the corresponding vector is the *cf-icf* product for the class c_i w.r.t p_j. Thus, the similarity between any two predicates is simply the cosine-similarity between the corresponding vectors.
- Baseline#2 augments Baseline#1 with context-based similarity. Thus, Baseline#2 computes the similarity between predicates as the average of the cosine-similarity and *CS*.

The accuracy values of the baselines in Table2 highlight the importance of *CS*. It is clear that on augmenting Baseline#1 with *CS*, its performance improves significantly. Even on its own, *CS* out-performs both the two baselines as well as the other similarity measures under consideration. This suggests *CS* models the semantic information as *contexts* and *shared contexts* effectively to compute semantic similarity between predicates.

Qualitative Evaluation

Results in Table3 quantify the quality of *PSS* on the DBpedia dataset. This evaluation attempts to examine the extent to which the human-perception of similarity resembles the similarity modeled by *PSS*. This task involves comparing the ranked-list (of sizes $1, 5, 10$) of similar predicates generated by *PSS* against that curated by experts as explained in Section5.2. Table 3 shows the F_N averaged scores across experts. It is observed that we obtain better F_N scores $@k = 10$ than $@k = 5$. This follows from the fact that $@\ k = 10$ the differences among the experts evens-out. The results $@k = 1$ indicate that the results of *PSS* were in agreement with experts 13 out of 15 times. This because result of each top-k list is evaluated by 3 experts and there are 5 predicates under evaluation. For P_5 $@k = 1$, the most-similar predicate we suggested matched with results of only one of one experts.

Thus, based on the results in Table3, we conclude *PSS* does a decent job at modelling similarity for predicates. Table4 compares the top-k results $@k = 10$ for dbo:draftTeam used in evaluation.

6. Conclusion

In this paper, we proposed a semantic similarity measure (*PSS*) exclusively for predicates in linked data. We proposed that *PSS* should be computed along two facets, namely *context* and *proximity*. To this end we introduced the context-based (*CS*) and proximity-based (*PS*) similarity measures. To facilitate evaluation, we constructed ground-truths for DBpedia, GeoSpecies and SWDF. Through experiments we show that *PSS* out-performs existing similarity measures. The results suggests that context-based measures enriched with capability to leverage relationships between predicates and classes are good at modelling similarity for predicates. Finally, the qualitative evaluation suggests that the *PSS* is effective in computing similarity between predicates in linked datasets.

References

[1] G. Zhu and C.A. Iglesias, Exploiting Semantic Similarity for Named Entity Disambiguation in Knowledge Graphs, *Expert Systems with Applications* (2018).

[2] L. Galárraga, G. Heitz, K. Murphy and F.M. Suchanek, Canonicalizing Open Knowledge Bases, in: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, ACM, 2014, pp. 1679–1688.

[3] K. Gunaratna, A.H. Yazdavar, K. Thirunarayan, A. Sheth and G. Cheng, Relatedness-based multi-entity summarization, in: *IJCAI: proceedings of the conference*, Vol. 2017, NIH Public Access, 2017, p. 1060.

[4] L. Fu, H. Wang, W. Jin and Y. Yu, Towards better understanding and utilizing relations in DBpedia, *Web Intelligence and Agent Systems: An International Journal* **10**(3) (2012), 291–303.

[5] Z. Zhang, A.L. Gentile, E. Blomqvist, I. Augenstein and F. Ciravegna, An Unsupervised Data-driven Method to Discover Equivalent Relations in Large Linked Datasets, *Semantic Web* (2015), 1–27.

Table 2

Pearsons Coefficients w.r.t to the gold-standard for several Semantic Similarity Measures (for predicates).

Sim. Measures	DBpedia	Geo Sp.	Sem Web DG
Wu Palmer (WUP)	0.18	-	-
Jaccard Sim.	0.15	0.61	0.54
Fu et al.	0.27	0.61	0.48
RDF2Vec (CBOW)	0.6	-	-
RDF2Vec (SG)	0.7	-	-
Baseline #1	0.4	0.24	-0.21
Baseline #2 (with CS)	0.43	0.83	0.57
CS (Context Sim. only)	0.76	0.81	0.75
PS (Proximity Sim. only)	0.54	-0.25	-0.06
PSS	**0.83**	**0.85**	**0.75**

Table 3

Qualitative Evaluation of PSS using Spearmans Footrule metric. The predicates in the random sample are P = {dbo:daylightSavingTimeZone, dbo:nationality, dbo:draftTeam, dbo:hubAirport, dbo:locationCity} We state F_N score averaged across the experts.

	P_1	P_2	P_3	P_4	P_5
$@k = 1$	1.0	1.0	1.0	1.0	0.33
$@k = 5$	0.66	0.56	0.6	0.66	0.6
$@k = 10$	0.802	0.778	0.7656	0.826	0.7131

Table 4

Qualitative comparison top-10 similar predicates for dbo:draftTeam generated by experts and by PSS

Expert #1	Expert #2	Expert #3	PSS Output
prospectTeam	prospectTeam	prospectTeam	prospectTeam
team	formerTeam	formerTeam	formerTeam
formerTeam	team	team	generalManager
generalManager	generalManager	nationalTeam	team
coach	almaMater	almaMater	sport
currentPartner	sport	coach	nationalTeam
almaMater	nationalTeam	generalManager	almaMater
nationalTeam	coach	currentPartner	coach
sport	currentPartner	sport	currentPartner
runningMate	runningMate	runningMate	runningMate

[6] H. Wang, Z. Fang, L. Zhang, J.Z. Pan and T. Ruan, Effective online knowledge graph fusion, in: *International Semantic Web Conference*, Springer, 2015, pp. 286–302.

[7] S. Likavec, F. Osborne and F. Cena, Property-based Semantic Similarity and Relatedness for Improving Recommendation Accuracy and Diversity, *International Journal on Semantic Web and Information Systems (IJSWIS)* **11**(4) (2015), 1–40.

[8] T. De Nies, C. Beecks, F. Godin, W. De Neve, G. Stepien, D. Arndt, L. De Vocht, R. Verborgh, T. Seidl, E. Mannens et al., Normalized Semantic Web Distance, in: *International Semantic Web Conference*, Springer, 2016, pp. 69–84.

[9] J. Sun, H. Qu, D. Chakrabarti and C. Faloutsos, Neighborhood formation and anomaly detection in bipartite graphs, in: *Data Mining, Fifth IEEE International Conference on*, IEEE, 2005, p. 8.

[10] Z. Wu and M. Palmer, Verbs semantics and lexical selection, in: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 1994, pp. 133–138.

[11] S. Harispe, S. Ranwez, S. Janaqi and J. Montmain, Semantic similarity from natural language and ontology analysis, *Synthesis Lectures on Human Language Technologies* **8**(1) (2015), 1–254.

[12] P. Ristoski and H. Paulheim, RDF2vec: RDF graph embeddings for data mining, in: *International Semantic Web Conference*, Springer, 2016, pp. 498–514.

[13] Y. Goldberg and O. Levy, word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method, *arXiv preprint arXiv:1402.3722* (2014).

[14] C. Schaefer, D. Hienert and T. Gottron, Normalized Relevance Distance-A Stable Metric for Computing Semantic Relatedness over Reference Corpora., in: *ECAI*, Vol. 263, 2014, pp. 789–794.

[15] R.L. Cilibrasi and P.M. Vitanyi, The google similarity distance, *IEEE Transactions on knowledge and data engineering* **19**(3) (2007).

[16] R.L. Cilibrasi and P. Vitanyi, Normalized web distance and word similarity, *arXiv preprint arXiv:0905.4039* (2009).

[17] R. Irny and P.S. Kumar, Mining inverse and symmetric axioms in Linked Data, in: *Joint International Semantic Technology Conference*, Springer, 2017, pp. 215–231.

[18] Z. Wu and M. Palmer, Verbs semantics and lexical selection, in: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 1994, pp. 133–138.

[19] R. Fagin, R. Kumar and D. Sivakumar, Comparing top k lists, *SIAM Journal on discrete mathematics* **17**(1) (2003), 134–160.