Computing the Semantic Similarity of Resources in DBpedia for Recommendation Purposes

Guangyuan Piao, Safina showkat Ara, and John G. Breslin

Insight Centre for Data Analytics
National University of Ireland Galway
IDA Business Park, Lower Dangan, Galway, Ireland
{guangyuan.piao,safina.ara}@insight-centre.org
{john.breslin}@nuigalway.ie

Abstract. The Linked Open Data cloud has been increasing in popularity, with DBpedia as a first-class citizen in this cloud that has been widely adopted across many applications. Measuring similarity between resources and identifying their relatedness could be used for various applications such as item-based recommender systems. To this end, several similarity measures such as *LDSD* (*Linked Data Semantic Distance*) were proposed. However, some fundamental axioms for similarity measures such as "equal self-similarity", "symmetry" or "minimality" are violated, and property similarities have been ignored. Moreover, none of the previous studies have provided a comparative study on other similarity measures. In this paper, we present a similarity measure, called *Resim* (*Resource Similarity*), based on top of a revised *LDSD* similarity measure. *Resim* aims to calculate the similarity of any resources in DBpedia by taking into account the similarity of the properties of these resources as well as satisfying the fundamental axioms. In addition, we evaluate our similarity measure with two state-of-the-art similarity measures (*LDSD* and *Shakti*) in terms of calculating the similarities for general resources (i.e., any resources without a domain restriction) in DBpedia and resources for music artist recommendations. Results show that our similarity measure can resolve some of the limitations of state-of-the-art similarity measures and performs better than them for calculating the similarities between general resources and music artist recommendations.

Keywords: Similarity measure, Recommender system, DBpedia

1 Introduction

The term Web of Data, often referred to as the Semantic Web, Web 3.0 or Linked Data, indicates a new generation of technologies responsible for the evolution of the current Web [10] from a Web of interlinked documents to a Web of interlinked data. The goal is to discover new knowledge and value from data, by publishing them using Web standards (primarily RDF [4]) and by enabling connections between heterogeneous datasets. In particular, the term Linked Open Data (LOD) denotes a set of best practices for publishing and linking structured data on the

Web. The project includes a great amount of RDF datasets interlinked with each other to form a giant global graph, which has been called the Linked Open Data cloud[1]. DBpedia[2] is a first-class citizen in the LOD cloud since it represents the nucleus of the entire LOD initiative [3]. It is the semantic representation of Wikipedia[3] and it has become one of the most important and interlinked datasets on the Web of Data. Compared to traditional taxonomies or lexical databases (e.g., WordNet [17]), it provides a larger and "fresher" set of terms, continuously updated by the Wikipedia community and integrated into the Web of Data. A resource in DBpedia represents any term/concept (e.g., Justin Bieber) as a dereferenceable URI (e.g., *http://dbpedia.org/resource/Justin_Bieber*) and provides additional information related to the resource. We use the prefix *dbpedia* for the namespace *http://dbpedia.org/resource/* in the rest of this paper. That is, *dbpedia:Justin_Bieber* denotes *http://dbpedia.org/resource/Justin_Bieber*.

On top of DBpedia, many approaches from different domains have been proposed by manipulating DBpedia resources and the relationships among them. For example, the resources can be used to represent a multi-domain user profile of interests across different Online Social Networks. In this case, an interest can be represented by a resource in DBpedia and the interest could be any topical resource that the user is interested in (e.g., *dbpedia:Justin_Bieber* or *dbpedia:Food*). Then, the user profile of interests can be used for personalization or recommendations [1, 2, 20]. It also has been widely adopted for improving the performance of recommender systems [6, 11, 18, 21, 22]. For instance, Heitmann et al. [11] proposed building open recommender systems which can utilize Linked Data to mitigate the *sparsity* problem of collaborative recommender systems [14].

Measuring similarity between resources and identifying their relatedness could be used for various applications, such as community detection in social networks or item-based recommender systems using Linked Data [23]. In this regard, several similarity measures were proposed for item-based recommendations [9, 13, 23, 24]. However, none of these studies evaluated over one or some of other similarity measures. Instead, each study proposed its own evaluation method for its measure. Hence, the performance compared to other similarity measures were not proven.

Secondly, despite different aspects of relatedness were considered in different similarity measures, property similarity is not incorporated within these measures. The Merriam-Webster Dictionary[4] defines property as "a special quality or characteristic of something". Also, from the definition of an ontology, the properties of each concept describe various features and attributes of the concept [19]. Thus, property similarity is important when there is no direct relationship between resources. For example, the similarity of two resources *dbpedia:Food* and *dbpedia:Fruit* using the *LDSD* (*Linked Data Semantic Distance*) similarity measure [23] (we will discuss the similarity measure in detail in Section 3.1)

[1] `http://lod-cloud.net/`
[2] `http://wiki.dbpedia.org/`
[3] `https://www.wikipedia.org/`
[4] `http://www.merriam-webster.com/`

is 0 since there is no direct link or shared resource via any properties. Similarly, the similarity of *dbpedia:Food* and *dbpedia:Rooster* is 0 as well. As a result, the recommender system cannot recommend any items (e.g., adverts) on *dbpedia:Fruit* or *dbpedia:Rooster* to the user if he or she is interested in the topic of *dbpedia:Food*. This can be addressed by considering shared incoming/outgoing properties since *dbpedia:Fruit* and *dbpedia:Food* have incoming properties such as *dbpedia-owl:industry* and *dbpedia-owl:product* in common (the prefix *dbpedia-owl* denotes the namespace *http://dbpedia.org/ontology/* in the rest of this paper).

Thirdly, some fundamental axioms [7] are violated for distance-based similarity measures such as:

- Equal self-similarity: $sim(A, A) = sim(B, B)$, for all stimuli A and B.
- Symmetry: $sim(A, B) = sim(B, A)$, for all stimuli A and B.
- Minimality: $sim(A, A) > sim(A, B)$, for all stimuli A $\neq$ B.

These are also common axioms for all word similarity measures in WordNet [16] and popular similarity measures based on graphs such as *SimRank* [12]. However, we found that the state-of-the-art similarity measures such as *LDSD* or *Shakti* [13] for calculating resource similarity do not satisfy at least two of these axioms (we will discuss this in detail in Section 2).

In this paper, we propose a new similarity measure named *Resim* (*Resource Similarity*), which is built on top of a revised *LDSD* and incorporates the similarity of properties. In this regard, *Resim* has two major components, one is a revised *LDSD* similarity measure to satisfy the three axioms mentioned above, and the other is a newly proposed property similarity measure. We choose *LDSD* since it works well in single-domain recommendations (see Section 4) and also has comparable results to supervised learning approaches [5, 6]. In addition, we compare and evaluate our similarity measure with *LDSD* and another state-of-the-art similarity measure named *Shakti*. These similarity measures were both devised for calculating the similarity between resources and recommendation purposes. To the best of our knowledge, this is the first comparative study of semantic similarity measures for DBpedia resources.

On top of that, we investigate if the performance of the item-based recommender system suffers from "*Linked Data sparsity*". Here, the *Linked Data sparsity problem* means that a lack of information on resources (e.g., small numbers of incoming/outgoing relationships from/to other resources) can decrease the performance of a recommender system.

The organization of the rest of the paper is as follows. In Section 2, we discuss related work on similarity measures of resources for recommendation purposes. In Section 3, we introduce our similarity measure - *Resim* - to calculate the similarity of resources in DBpedia. Section 4 elaborates on the experimental setup for the evaluation of our similarity measure with others, and Section 5 highlights the results. In Section 6, we study if Linked Data sparsity has an effect on the performance of the item-based recommender system that adopts the similarity measure for calculating the similarity between items (resources in DBpedia in this paper). Finally, Section 7 concludes the paper and gives some ideas for future work.

$$LDSD(r_a, r_b) = \frac{1}{1 + \sum_i \frac{C_d(l_i, r_a, r_b)}{1+\log(C_d(l_i, r_a, n))} + \sum_i \frac{C_d(l_i, r_b, r_a)}{1+\log(C_d(l_i, r_b, n))} + \sum_i \frac{C_{ii}(l_i, r_a, r_b)}{1+\log(C_{ii}(l_i, r_a, n))} + \sum_i \frac{C_{io}(l_i, r_a, r_b)}{1+\log(C_{io}(l_i, r_a, n))}} \quad (1)$$

$$LDSD_{sim}(r_a, r_b) = 1 - LDSD(r_a, r_b) \quad (2)$$

2 Related Work

Maedche et. al [15] defined a set of similarity measures for comparing ontology-based metadata by considering different aspects of an ontology separately. They propose differentiating across three dimensions for comparing two resources: taxonomic, relational and attribute similarities. However, the similarity measures depend on some strong assumptions about the model such as "Ontologies are strictly hierarchical such that each concept is subsumed by only one concept", which is not the case in terms of DBpedia.

Passant [23] proposed a measure named *LDSD* to calculate semantic distance on Linked Data. The distance measure (equation (1)) considers direct links from resource A to resource B and vice versa (C_d, C_{ii} and C_{io} functions are detailed in Section 3.1). In addition, it also considers the same incoming and outgoing nodes via the same properties of resources A and B in a graph (an example is given in Fig. 1). The distance measure has a scale from 0 to 1, where a larger value denotes less similarity between two resources. Thus, the similarity measure can be defined using equation (2), and we will use $LDSD_{sim}$ to denote the similarity measure in the rest of the paper. In later work, the author used the *LDSD* similarity measure in a recommender system based on DBpedia resources which recommends similar music artists based on the artists in a user's preference profile [22].

While $LDSD_{sim}$ works well in single-domain recommendations, there are several problems that need to be addressed. Since the measure is based on a count of direct/indirect links for resources, a higher number of these relationships can lead to higher similarity. However, the similarity would never be 1 even for the same resources, which leads to "different self-similarity". That is, $sim(r_a, r_a)$ and $sim(r_b, r_b)$ will be different even though both are close to 1. For instance, the similarity of *dbpedia:Doctor* with itself is 0.967 while the similarity of *dbpedia:Professor* with itself is 0.998. Also, the measure produces non-symmetric results for $sim(r_a, r_b)$ and $sim(r_b, r_a)$ (We will discuss this in detail in Section 3). Furthermore, it fails to calculate the similarities on general resource pairs (e.g., any resource used in DBpedia for representing a user's interests). For instance, both *sim*(*dbpedia:Doctor*, *dbpedia:Professor*) and *sim*(*dbpedia:Doctor*, *dbpedia:Cucumber*) will be 0, thus we cannot recommend items on some similar topics if *dbpedia:Doctor* is one of the interests of a user.

Leal et al. [13] presents an approach for computing the semantic relatedness of resources in DBpedia. In the paper, they proposed a similarity measure based

on a notion of proximity, which measures how connected two resources are, rather than how distant they are. This means that the similarity measure considers both distance and the number of paths between two nodes. The similarity measure extends each step to find longer paths between two resources and penalizes proximity by steps, i.e., a longer path contributes less to the proximity and the extension is terminated by a defined value of maximum steps (max step). The similarity measure is implemented in a tool named "Shakti", which extracts an ontology for a given domain from DBpedia and uses it to compute the semantic relatedness of resources. We use *Shakti* to refer to this measure in the rest of the paper. However, they do not consider incoming nodes (resources) and properties of the resources as $LDSD_{sim}$ did. Furthermore, the proximity value for the same resources would be 0 since they will be removed before any extension. As a result, $sim(r_a, r_b) > sim(r_a, r_a)$ and thus violates the "minimality" axiom. In addition, the weights assigned to properties are defined manually and the authors pointed out the need for a sounder (automated) approach as future work.

Based on the *Shakti* measure, Strobin et al. [24] propose a method to find the weights automatically by using a genetic optimization algorithm based on a training dataset from Last.fm[5]. This method is quite efficient at learning the weights automatically. However, it needs a gold standard dataset (e.g., Last.fm dataset for music domain) to learn the weights of properties which is not always available in other domains.

For evaluation, every work proposes its own evaluation method for its measure and none of these studies have compared their proposed similarity measures to others. For example, some have evaluated the similarity measures in terms of specific domains of recommender systems [9, 13, 22, 23] while others have evaluated them in terms of clustering problems [15].

In this work, we propose our similarity measure and also provide a comparative evaluation over $LDSD_{sim}$ and *Shakti* in terms of calculating the similarity of general resources (i.e., any resource in DBpedia without a domain restriction) and single-domain recommendations to examine the pros and cons of each measure.

3 Resim Similarity Measure

In this section, we present a similarity measure named *Resim* (*Resource Similarity*) to calculate the similarity of resources in DBpedia. The method is built on top of the $LDSD_{sim}$ measure and resolve its aforementioned limitations. In this regard, we first discuss each component of $LDSD_{sim}$ in Section 3.1 and elaborate upon their limitations. Then we describe the components of *Resim* that resolve these limitations and also satisfy the axioms of "equal self-similarity", "minimality" (Section 3.2) and "symmetry" (Section 3.3). In addition, we present a method to calculate the property similarity of resources in Section 3.4 and present a final equation for *Resim* in Section 3.5. We use the definition of a

[5] `http://last.fm`

dataset following the Linked Data principles outlined in [22].

Definition 1. A dataset following the Linked Data principles is a graph G such as $G = (R, L, I)$ in which $R = \{r_1, r_2, ...r_n\}$ is a set of resources identified by their URI, $L = \{l_1, l_2, ...l_n\}$ is a set of typed links identified by their URI and $I = \{i_1, i_2, ...i_n\}$ is a set of instances of these links between resources, such as $i_i =< l_j, r_a, r_b >$.

3.1 LDSD similarity measure

The $LDSD_{sim}$ measure (see equations (1) and (2)) consists of two C_d functions with $C_{ii}(l_i, r_a, r_b)$ and $C_{io}(l_i, r_a, r_b)$. C_d is a function that computes the number of direct and distinct links between resources in a graph G. $C_d(l_i, r_a, r_b)$ equals 1 if there is an instance of l_i from resource r_a to resource r_b. Otherwise, if there is no instance of l_i from resource r_a to resource r_b, $C_d(l_i, r_a, r_b)$ equals to 0. By extension C_d can be the total number of distinct instances of the link l_i from r_a to any node ($C_d(li, r_a, n)$). For example, in the example graph (Fig. 1), we have:

$C_d(influences, Ariana\_Grande, Selena\_Gomez) = 1$
$C_d(influences, Ariana\_Grande, n) = 1$
$C_d(musicalguests, List\_of\_The\_Tonight\_Show\_with\_Jay\_Leno\_episodes\_(2013–14), n) = 2$

C_{ii} and C_{io} are functions that compute the number of indirect and distinct links, both incoming and outgoing, between resources in a graph G. $C_{ii}(l_i, r_a, r_b)$ equals 1 if there is a resource n that satisfy both $< l_i, r_a, n >$ and $< l_i, r_b, n >$, 0 if not. Similarly, $C_{io}(l_i, r_a, r_b)$ equals 1 if there is a resource n that is linked to both r_a and r_b via outgoing l_i, 0 if not. In the example (Fig. 1), we have $C_{ii}(musicalguests, Ariana\_Grande, Selena\_Gomez) = 1$ (via incoming property from *List_of_The_Tonight_Show_with_Jay_Leno_episodes_(2013-14)*) and $C_{io}(subject, Ariana\_Grande, Selena\_Gomez) = 1$ (via outgoing property to *Category:21st-century_American_singers*).

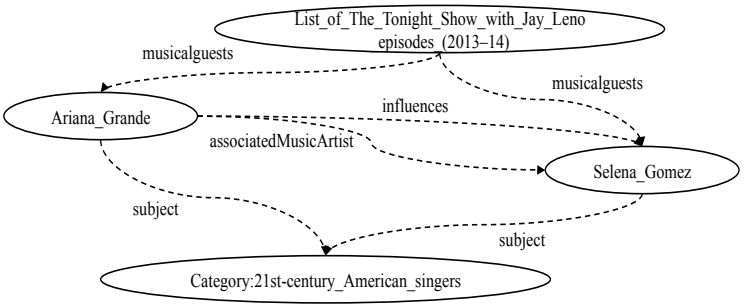

Fig. 1. Example of relationships of two resources in DBpedia

$$LDSD'(r_a, r_b) = \frac{1}{1 + \sum_i \frac{C_d(l_i, r_a, r_b)}{1+\log(C_d(l_i, r_a, n))} + \sum_i \frac{C_d(l_i, r_b, r_a)}{1+\log(C_d(l_i, r_b, n))} + \sum_i \frac{C_{ii}(l_i, r_a, r_b)}{1+\log(\frac{C_{ii}(l_i, r_a, n)+C_{ii}(l_i, r_b, n)}{2})} + \sum_i \frac{C_{io}(l_i, r_a, r_b)}{1+\log(\frac{C_{io}(l_i, r_a, n)+C_{io}(l_i, r_b, n)}{2})}} \quad (4)$$

$$LDSD'_{sim}(r_a, r_b) = 1 - LDSD'(r_a, r_b) \quad (5)$$

3.2 Equal self-similarity and minimality

"Equal self-similarity" denotes that the similarity of the same resources should be the same, while "minimality" denotes that the similarity of the same resources should be bigger than the similarity of two different resources. Formally, these axioms can be defined as below [7]:

- Equal self-similarity: $sim(r_a, r_a) = sim(r_b, r_b)$, for all resources r_a and r_b.
- Minimality: $sim(r_a, r_a) > sim(r_a, r_b)$, for all resources $r_a \neq r_b$.

In order to achieve "equal self-similarity", we can simply define conditions such as "the similarity of two resources r_a and r_b equals 1 if the two resources are exactly the same or r_a and r_b have the *owl:sameAs* relationship". Such an *owl:sameAs* statement indicates that two resources in DBpedia refer to the same thing. Hence, the first component of our method can be defined as below:

$$Resim(r_a, r_b) = 1, \text{ if } URI(r_a) = URI(r_b) \text{ or } r_a \ owl{:}sameAs \ r_b \quad (3)$$

The similarity measure thus scales from 0 to 1, and the similarity of r_a and r_b will be 1 if the two resources are exactly the same or if r_a and r_b have the *owl:sameAs* relationship. Otherwise, the similarity of two resources will be less than 1. As a result, the similarity measure satisfies both "equal self-similarity" and "minimality" axioms.

3.3 Symmetry

The "symmetry" axiom denotes that the similarity of two resources r_a and r_b will be the same as the similarity for a reversed order of the two resources. Formally, it can be defined as:

- Symmetry: $sim(r_a, r_b) = sim(r_b, r_a)$, for all resources r_a and r_b.

As we can see from equation (1), the sum of two C_d functions produces the same results for the similarities of two resources and the reversed order of them. That is, $LDSD_{sim}(r_a, r_b) = LDSD_{sim}(r_b, r_a)$ while considering C_d functions only. The non-symmetric results occur due to the normalization parts of C_{ii}

and C_{io} functions. The normalizations of the two functions are carried out using the logarithmic value of all incoming/outgoing nodes of r_a for $LDSD_{sim}(r_a, r_b)$. However, when calculating the similarity of resources with a reversed order - $LDSD_{sim}(r_b, r_a)$, the normalizations of C_{ii} and C_{io} are carried out using the logarithmic value of all incoming/outgoing nodes of r_b. This means that the different normalizations used when reversing the order of r_a and r_b causes non-symmetric results. Thus, we modify $LDSD$ as $LDSD'$ in equation (4) and $LDSD_{sim}$ as $LDSD'_{sim}$ in equation (5). The modified normalization considers incoming/outgoing nodes of both r_a and r_b. Hence, the similarities for two resources and their reversed order are the same, so this satisfies the "symmetry" axiom.

3.4 Property similarity

Using the aforementioned definitions of property in Section 1, we add the property similarity for resources as an additional component to *Resim*. This property similarity is defined in equation (6). Our intuition behind this is that the property similarity of resources is important when the relationship of two resources is not available.

$$Property_{sim}(r_a, r_b) = \frac{\sum_i \frac{C_{sip}(l_i, r_a, r_b)}{C_d(l_i, n, n)}}{C_{ip}(r_a) + C_{ip}(r_b)} + \frac{\sum_i \frac{C_{sop}(l_i, r_a, r_b)}{C_d(l_i, n, n)}}{C_{op}(r_a) + C_{op}(r_b)} \tag{6}$$

Definition 2. C_{sip} and C_{sop} are functions that compute the number of distinct shared incoming and outgoing links (properties) between resources in a graph G. $C_{sip}(l_i, r_a, r_b)$ equals 1 if there is an incoming link l_i that exists for both r_a and r_b, and $C_{sop}(l_i, r_a, r_b)$ equals 1 if there is an outgoing link l_i that exists for both r_a and r_b. C_{ip} and C_{op} are functions that compute the number of incoming and outgoing links for a resource. $C_d(l_i, n, n)$ (see Section 3.1) denotes the total number of distinct instances of the link l_i between any two resources.

One thing to note is that we normalize the weight of C_{sip} (C_{sop}) by the total number of distinct instances of the link l_i between any two resources in G instead of the logarithm of the number. This will penalize frequently appearing properties more heavily, and we found that this approach as a part of *Resim* yields a better result for recommendations than the logarithm value of the number.

3.5 Resim similarity measure

Based on the components discussed above, *Resim* combines these components by calculating the weighted arithmetic mean of $Property_{sim}$ and $LDSD'_{sim}$. The final equation for *Resim* is defined as follows:

$$Resim(r_a, r_b) = \begin{cases} 1, & \text{if } URI(r_a) = URI(r_b) \text{ or } r_a \textit{ owl:sameAs } r_b \\ \frac{w_1 * Property_{sim}(r_a, r_b) + w_2 * LDSD'_{sim}(r_a, r_b)}{w_1 + w_2}, & \text{otherwise} \end{cases} \tag{7}$$

The weights may be adjusted according to the given dataset for which the measures should be applied (e.g. within our empirical evaluation we used a weight of 1 for w_1 and 2 for w_2 to give higher importance to the relationships between two resources).

4 Evaluation Setup

In this section, we describe an experiment to evaluate our similarity measure compared to $LDSD_{sim}$ and *Shakti*. For the *Shakti* similarity measure, we use the weights of properties manually assigned by the authors in [13]. In *Shakti*, seven properties related to the music domain have been considered such as *dbpedia-owl:genre, instrument, influences, associatedMusicalArtist, associatedBand, currentMember* and *pastMember*.

Firstly, we examine these similarity measures in terms of the three axioms: "equal self-similarity", "symmetry" and "minimality".

Secondly, we aim to evaluate the performance of calculating similarities on general resources without restricting to any domain, i.e., for any resources in DBpedia. For example, the similarity of the two resources *dbpedia:Cat* and *dbpedia:Dog* should be higher than that of *dbpedia:Cat* and *dbpedia:Human*, and a test pair can be created as *sim*(*dbpedia:Cat*, *dbpedia:Dog*) > *sim*(*dbpedia:Cat*, *dbpedia:Human*). In order to get the gold standard test pairs, we use the WordSim353 dataset [8]. WordSim353 is a dataset containing English word pairs along with human-assigned similarity judgements on a scale from 0 (totally unrelated words) to 10 (very much related or identical words), and is used to train and/or test algorithms implementing semantic similarity measures (i.e., algorithms that numerically estimate the similarity of natural language words). We retrieved word pairs from the dataset that satisfy $sim(W_a, W_b) > sim(W_a, W_c)$ where the difference is higher than 2. For instance, the word "car" appears several times with words such as "automobile" and "flight" among the word pairs, and $sim(car, automobile) = 8.49 > sim(car, flight) = 4.94$. We then retrieve the corresponding DBpedia resources (i.e., *dbpedia:Car*, *dbpedia:Automobile*, *dbpedia:Flight*) and construct a test pair as *sim*(*dbpedia:Car*, *dbpedia:Automobile*) > *sim*(*dbpedia:Car*, *dbpedia:Flight*). In all, 28 test pairs of resources were retrieved (see Table 2). We evaluate the similarity measures on these test cases and see how many of them can be satisfied by each similarity measure.

Table 1. Similarity measures evaluated on axioms

Axiom	LDSDsim	Shakti	Resim
Equal self-similarity			√
Symmetry		√	√
Minimality	√		√

Table 2. Evaluation on test pairs of resources based on extracted word pairs from WordSim353

Test pairs of resources			**LDSDsim**	**Shakti**	**Resim**
sim(dbpedia:Car, dbpedia:Automobile)	>	sim(dbpedia:Car, dbpedia:Flight)	√	√	√
sim(dbpedia:Money, dbpedia:Currency)	>	sim(dbpedia:Money, dbpedia:Business_operations)		√	√
sim(dbpedia:Money, dbpedia:Cash)	>	sim(dbpedia:Money, dbpedia:Bank)			
sim(dbpedia:Money, dbpedia:Cash)	>	sim(dbpedia:Money, dbpedia:Demand_deposit)	√		√
sim(dbpedia:Professor, dbpedia:Doctor_of_Medicine)	>	sim(dbpedia:Professor, dbpedia:Cucumber)	√	√	√
sim(dbpedia:Doctor_of_Medicine, dbpedia:Nursing)	>	sim(dbpedia:Doctor_of_Medicine, dbpedia:Bus_driver)		√	√
sim(dbpedia:Ocean, dbpedia:Sea)	>	sim(dbpedia:Ocean, dbpedia:Continent)	√	√	√
sim(dbpedia:Computer, dbpedia:Keyboard)	>	sim(dbpedia:Computer, dbpedia:News)			
sim(dbpedia:Computer, dbpedia:Internet)	>	sim(dbpedia:Computer, dbpedia:News)	√	√	√
sim(dbpedia:Computer, dbpedia:Software)	>	sim(dbpedia:Computer, dbpedia:Laboratory)	√	√	√
sim(dbpedia:Cup, dbpedia:Drink)	>	sim(dbpedia:Cup, dbpedia:Article)		√	√
sim(dbpedia:Cup, dbpedia:Coffee)	>	sim(dbpedia:Cup, dbpedia:Substance)		√	√
sim(dbpedia:Drink, dbpedia:Mouth)	>	sim(dbpedia:Drink, dbpedia:Ear)			√
sim(dbpedia:Drink, dbpedia:Eating)	>	sim(dbpedia:Drink, dbpedia:Mother)	√		√
sim(dbpedia:Football, dbpedia:Association_football)	>	sim(dbpedia:Football, dbpedia:Basketball)		√	
sim(dbpedia:Monarch, dbpedia:Queen_consort)	>	sim(dbpedia:Monarch, dbpedia:Cabbage)	√	√	√
sim(dbpedia:Tiger, dbpedia:Jaguar)	>	sim(dbpedia:Tiger, dbpedia:Organism)	√	√	√
sim(dbpedia:Day, dbpedia:Night)	>	sim(dbpedia:Day, dbpedia:Summer)		√	√
sim(dbpedia:Coast, dbpedia:Shore)	>	sim(dbpedia:Coast, dbpedia:Forest)	√		√
sim(dbpedia:Coast, dbpedia:Shore)	>	sim(dbpedia:Coast, dbpedia:Hill)	√		√
sim(dbpedia:Governor, dbpedia:Office)	>	sim(dbpedia:Governor, dbpedia:Interview)			
sim(dbpedia:Food, dbpedia:Fruit)	>	sim(dbpedia:Food, dbpedia:Rooster)			√
sim(dbpedia:Life, dbpedia:Death)	>	sim(dbpedia:Life, dbpedia:Term_(time))	√	√	√
sim(dbpedia:Digital_media, dbpedia:Radio)	>	sim(dbpedia:Digital_media, dbpedia:Trade)	√	√	√
sim(dbpedia:Planet, dbpedia:Moon)	>	sim(dbpedia:Planet, dbpedia:People)		√	√
sim(dbpedia:Opera, dbpedia:Performance)	>	sim(dbpedia:Opera, dbpedia:Industry)		√	√
sim(dbpedia:Nature, dbpedia:Environment)	>	sim(dbpedia:Nature, dbpedia:Man)		√	
sim(dbpedia:Energy, dbpedia:Laboratory)	>	sim(dbpedia:Energy, dbpedia:Secretary)			√
		Total :	13	18	23

Finally, we evaluate the similarity measure by adopting it to item-based recommendations in the music domain. The recommender system recommends the top-N similar music artists for a music artist based on the similarities among all candidates and the music artist. Passant [22] evaluated the *LDSD* measure in the music domain by comparing with a recommendations list from Last.fm. Last.fm offers a ranked list of similar artists/bands for each artist/band based on their similarities. They showed that in spite of a slight advantage for Last.fm, *LDSD* based recommendations achieved a reasonable score, especially considering that it does not use any collaborative filtering approach, and relies only on links between resources. Similarly, we adopt the recommendations list from Last.fm to evaluate the performance of the recommendations. First of all, all the resources of type of *dbpedia-owl:MusicArtist* or *dbpedia-owl:Band* are extracted via the DBpedia SPARQL endpoint[6]. By doing so, 75,682 resources are obtained consisting of 45,104 resources of type *dbpedia-owl:MusicArtist* and 30,578 resources of type *dbpedia-owl:Band*. Then we randomly selected 10 resources from these 75,682 resources. For each resource (a music artist or band in this case), we manually get the top 10 recommendations list from Last.fm for each resource which can be found in DBpedia. To construct a candidate list for recommendations, we create a candidate list with these top 10 recommendations from Last.fm and 200 randomly selected resources among the 75,682 resources of type *dbpedia-owl:MusicArtist* or *dbpedia-owl:Band*. For example, if a user is interested in the music artist *dbpedia:Ariana_Grande*, the candidate list consists of the top 10 similar music artists recommended by Last.fm (that can be found in DBpedia) and 200 randomly selected resources of type *dbpedia-owl:MusicArtist* or *dbpedia-owl:Band*. Then we calculate the similarities between *dbpedia:Ariana_Grande* and the candidate list with *Resim*, $LDSD_{sim}$ and *Shakti* to get the top-N recommendations. Our goal is to see the performance of the top-N recommendations based on these similarity measures.

The performance of the recommendations was measured by means of R@N and MRR (Mean Reciprocal Rank). For a resource (e.g., *dbpedia:Ariana_Grande*), recall at N (R@N) is the fraction of resources that are relevant to the resource that are successfully retrieved in the top-N recommendations and MRR indicates at which rank the first item relevant to the resource occurs on average.

We use N = 5, 10 and 20 in the evaluation and report the results of averaged R@N over the 10 randomly selected resources of *dbpedia-owl:MusicArtist* or *dbpedia-owl:Band* based on *Resim*, $LDSD_{sim}$, *Shakti*. Since the *Shakti* similarity measure uses the value of max step for the extension of the paths between two resources, we use 3 and 5 for the value of max step and denote these variants as *Shakti3* (*max step set to 3*) and *Shakti5* (*max step set to 5*).

[6] `http://dbpedia.org/sparql`

5 Results

Table 1 shows the details of three similarity measures on the three axioms: "equal self-similarity", "symmetry" and "minimality". As we can see from the table, *Resim* satisfies all of the axioms while both $LDSD_{sim}$ and *Shakti* do not satisfy two of the axioms.

The details of results for test pairs of general resources are presented in Table 2. Among these test cases of general resources, *Resim* can correctly calculate the similarity of resources and satisfy 23 test pairs out of 28 on account of $Property_{sim}$, while $LDSD_{sim}$ and *Shakti* can satisfy 13 and 18 pairs respectively. In more detail, $LDSD_{sim}$ failed to calculate the similarities for many of these general resource pairs. That is, $sim(r_a, r_b) = 0$ since there was no relationship between them. In this case, a recommender system based on $LDSD_{sim}$ cannot recommend anything to a user. For instance, a user has an interest on the topic of *dbpedia:Money* and there are two news items on the topics of *dbpedia:Currency* and *dbpedia:Business_operations*. Based on $LDSD_{sim}$, the recommender system cannot recommend any news for these topics to the user since both $LDSD_{sim}(dbpedia:Money, dbpedia:Currency)$ and $LDSD_{sim}(dbpedia:Money, dbpedia:Business\_operations)$ are 0. From the results of the *Shakti* similarity measure, incorporating the number of paths between two resources improved the performance. However, it also generates some incorrect results by incorporating the number of paths in some cases such as $sim(dbpedia:Money, dbpedia:Cash)$ and $sim(dbpedia:Money, dbpedia:Demand\_deposit)$.

The results of R@N and MRR for recommendations based on the randomly selected music artists and bands are displayed in Fig. 2. The *paired t-test* is used for testing the significance where the significance level was set to 0.05 unless otherwise noted. Overall, *Resim* performed better than $LDSD_{sim}$, *Shakti3* and *Shakti5* and achieved 48% recall for the top 10 recommendations. In more detail, both $LDSD_{sim}$ and *Resim* outperform *Shakti3* and *Shakti5* significantly in terms of R@N and MRR. In addition, the results of R@N are increased by 2% and 1% respectively when n is equal to 5 and 10 using *Resim* compared to

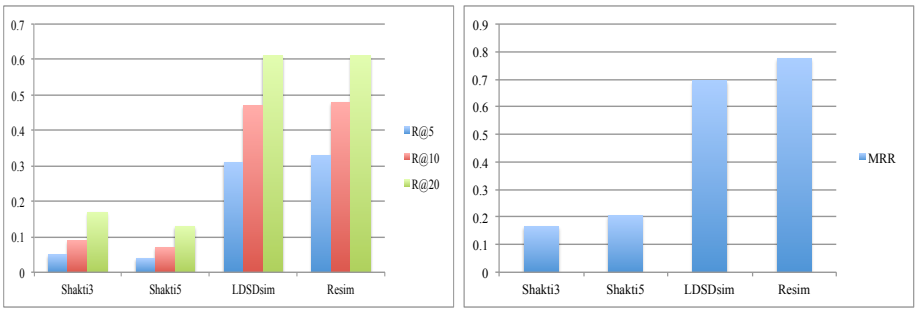

Fig. 2. Average recall at n and MRR for the recommendations of 10 random samples

the results using $LDSD_{sim}$. The result of MRR is increased by 8% using *Resim* compare to the result using $LDSD_{sim}$. One thing to note in our experiment is that the higher value of max step for the *Shakti* similarity measure did not improve the recall. Conversely, the performance in terms of R@N is decreased by incorporating more steps in the *Shakti* similarity measure.

To summarise, *Resim* satisfies the axioms as a similarity measure and performs better at calculating the similarities of general resources, compared to the $LDSD_{sim}$ and *Shakti* similarity measures. For single-domain resources, *Resim* has a similar but slightly better performance compared to $LDSD_{sim}$ and significantly better performance than *Shakti*.

6 Study of Linked Data Sparsity Problem

During the experiment mentioned in the previous section, we found that some of the random samples with less incoming/outgoing links yielded poor recall. For instance, the recall at 10 of recommendations for *dbpedia:Jasmin_Thompson* is 0.1, which is one of the random samples that has 42 outgoing links and 3 incoming links. In contrast, the recall of recommendations for the *dbpedia:Dead_Kennedys* is 0.9, which has 117 outgoing links and 119 incoming links.

This observation motivates us to investigate if the performance of the item-based recommender system suffers from "*Linked Data sparsity*". Here, the *Linked Data sparsity problem* means that the performance of the recommender system based on similarity measures of resources decreases when resources lack information (i.e., when they have a lesser number of incoming/outgoing relationships to other resources). In this regard, the null hypothesis to test can be defined as below:

H_0 : *The number(log) of incoming/outgoing links for resources has no relationship to the performance of a recommender system.*

We use the logarithm of the number (denote as number(log)) to decrease the variation in numbers. We reject the null hypothesis if the number(log) of incoming/outgoing links and the recall of recommendations have a strong relationship (*Pearson's correlation* > 0.4), otherwise we accept the null hypothesis.

To this end, we additionally selected 10 popular DBpedia resources of type *dbpedia-owl:MusicArtist* as samples, and then calculate the recall at 5, 10 and 20 in the same way as we did for the 10 randomly selected samples. The assumption here is that the popular samples tend to have more information (i.e., incoming/outgoing links) than random samples. This is because these resources in DBpedia are a reflection of the corresponding concepts/articles in Wikipedia, and usually popular music artists have more information thanks to a higher number of contributors.

First, we intend to see if the recommendation system performs better on popular samples than on random ones. On top of that, we aim to investigate the

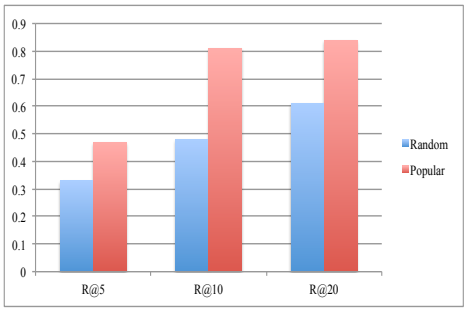

Fig. 3. Recall of recommendations on random samples and popular ones

correlation by calculating the *Pearson's coefficient* between the number(log) of incoming/outgoing links for resources and the recall of the recommender system.

As we can see from Fig. 3, the recall results of the recommender system on popular samples are significantly better than the results on random samples. Following this finding, we calculate the correlation between the number(log) of incoming/outgoing links for resources and the performance (recall) of the recommender system. We report R@10 based on *Resim* here, and similar results can be observed by using other measures. The result shows the performance of the recommender system has a very strong positive relationship (Fig. 4, *Pearson's correlation of 0.798*) with the total number(log) of incoming/outgoing links ($p < 0.01$). Hence, the null hypothesis is rejected. In other words, the performance of the recommender system decreases for the resources with sparsity (i.e., less incoming/outgoing links). It also indicates that, on one hand, utilizing Linked Data to build a recommender system can mitigate the traditional sparsity problem [11] of collaborative recommender systems, but on the other hand, the system can also have a *Linked Data sparsity problem* for resources in the Linked Data set that the recommender system has adopted.

7 Conclusion and Future Work

In this paper, we introduced a similarity measure called *Resim* (*Resource Similarity*) to calculate semantic similarity for resources in DBpedia. Based on the work of $LDSD_{sim}$, we tackled some of the limitations of this similarity measure and constructed our similarity measure to resolve these limitations so as to satisfy some fundamental axioms. In addition, we incorporated property similarity in *Resim* to calculate the similarity of two resources when the relationship between them is not available. An evaluation on test pairs of general resources shows that incorporating property similarity can improve the performance of calculating similarities for general resources. Furthermore, an evaluation based on the top n recommendations in the music domain shows that our similarity measure outperforms $LDSD_{sim}$ and significantly improves performance over the *Shakti* similarity measure. In addition, we investigated if the performance of an

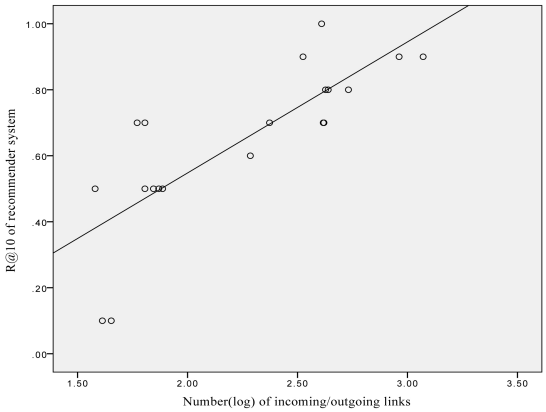

Fig. 4. Scatter plot of R@10 and number(log) of links, r=0.798

item-based recommender system, which adopts similarity measures for calculating the similarity between items (resources), suffers from the "*Linked Data sparsity problem*", and proved that the performance of the recommender system has a very strong positive relationship with the number(log) of the total number of incoming/outgoing links ($p < 0.01$) for resources.

In future work, we propose to extend the current similarity measure by incorporating longer paths, while being mindful that a trade off between performance and accuracy might be a challenge. In addition, we plan to extend our similarity measure by incorporating paths to calculate the similarity of a user interest graph which can then be applied to social recommender systems.

Acknowledgments. This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289 (Insight Centre for Data Analytics).

References

1. Abel, F., Gao, Q., Houben, G.J., Tao, K.: Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In: Proceedings of the 3rd International Web Science Conference. p. 2. ACM (2011)
2. Abel, F., Herder, E., Houben, G.J., Henze, N., Krause, D.: Cross-system user modeling and personalization on the social web. User Modeling and User-Adapted Interaction 23(2-3), 169–209 (2013)
3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. Springer (2007)
4. Brickley, D., Guha, R.V.: {RDF vocabulary description language 1.0: RDF schema} (2004)
5. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D.: Exploiting the web of data in model-based recommender systems. In: Proceedings of the sixth ACM conference on Recommender systems. pp. 253–256. ACM (2012)

6. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D., Zanker, M.: Linked Open Data to Support Content-based Recommender Systems. In: Proceedings of the 8th International Conference on Semantic Systems. pp. 1–8. I-SEMANTICS '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2362499.2362501
7. Ennis, F.G.A., M., D.: Similarity Measures (2007), http://www.scholarpedia.org/article/Similarity_measures
8. Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: The concept revisited. In: Proceedings of the 10th international conference on World Wide Web. pp. 406–414. ACM (2001)
9. Groues, V., Naudet, Y., Kao, O.: Adaptation and evaluation of a semantic similarity measure for dbpedia: A first experiment. In: Semantic and Social Media Adaptation and Personalization (SMAP), 2012 Seventh International Workshop on. pp. 87–91. IEEE (2012)
10. Heath, T., Bizer, C.: Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology 1(1), 1–136 (2011)
11. Heitmann, B., Hayes, C.: Using Linked Data to Build Open, Collaborative Recommender Systems. In: AAAI spring symposium: linked data meets artificial intelligence. pp. 76–81 (2010)
12. Jeh, G., Widom, J.: SimRank: A Measure of Structural-context Similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 538–543. KDD '02, ACM, New York, NY, USA (2002)
13. Leal, J.P., Rodrigues, V., Queirós, R.: Computing semantic relatedness using dbpedia (2012)
14. Lee, S., Yang, J., Park, S.Y.: Discovery of hidden similarity on collaborative filtering to overcome sparsity problem. In: Discovery Science. pp. 396–402. Springer (2004)
15. Maedche, A., Zacharias, V.: Clustering ontology-based metadata in the semantic web. In: Principles of Data Mining and Knowledge Discovery, pp. 348–360. Springer (2002)
16. Meng, L., Huang, R., Gu, J.: A review of semantic similarity measures in wordnet. International Journal of Hybrid Information Technology 6(1), 1–12 (2013)
17. Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM 38(11), 39–41 (1995)
18. Musto, C., Basile, P., Lops, P., de Gemmis, M., Semeraro, G.: Linked Open Data-enabled Strategies for Top-N Recommendations. CBRecSys 2014 p. 49 (2014)
19. Noy, N.F., McGuinness, D.L.: Ontology development 101: A guide to creating your first ontology (2001)
20. Orlandi, F., Breslin, J., Passant, A.: Aggregated, interoperable and multi-domain user profiles for the social web (2012)
21. Ostuni, V.C., Di Noia, T., Di Sciascio, E., Mirizzi, R.: Top-n recommendations from implicit feedback leveraging linked open data. In: Proceedings of the 7th ACM conference on Recommender systems. pp. 85–92. ACM (2013)
22. Passant, A.: dbrec: Music Recommendations Using DBpedia. In: ISWC 2010 SE - 14. pp. 209–224 (2010)
23. Passant, A.: Measuring Semantic Distance on Linking Data and Using it for Resources Recommendations. In: AAAI Spring Symposium: Linked Data Meets Artificial Intelligence. vol. 77, p. 123 (2010)
24. Strobin, L., Niewiadomski, A.: Evaluating semantic similarity with a new method of path analysis in RDF using genetic algorithms. COMPUTER SCIENCE 21(2), 137–152 (2013)