Measuring Semantic Distance on Linking Data and Using it for Resources Recommendations *

Alexandre Passant
Digital Enterprise Research Institute
National University of Ireland, Galway
IDA Business Park, Lower Dangan, Galway, Ireland
alexandre.passant@deri.org

Abstract

A frequent topic discussed in the Linked Data community, especially when trying to outreach its values, is *"What can we do with all this data ?"*. In this paper, we demonstrate (1) how to measure semantic distance on Linked Data in order to identify relatedness between resources, and (2) how such measures can be used to provide a new kind of self-explanatory recommendations, bringing together Linked Data and Artificial Intelligence principles, and demonstrating how intelligent agents could emerge in the realm of Linked Data.

Introduction

So far, the Linking Open Data (LOD) initiative[1] has been quite successful in terms of publishing and interlinking data on the Web: from a few datasets two years ago, the LOD Cloud now features almost 100 of them (Table 1[2]), including user-generated content (Freebase), data from public companies (BBC) as well as bio-medical information (LODD[3]). Thanks to it, the Web of Data is now a reality.

However, a question that often arises in related discussions concerns how we can efficiently take advantage of it (Heath 2008). While we can argue that the most important feature of the LOD initiative is to provide *"raw data now"*[4], building applications on the top of this data could provide more incentives to developers / CEOs / CTOs to publish their content as such. This would consequently lead to a virtuous circle of producing and consuming Linked Data, enriching the value of this global network, by analogy with Metcalfe's law (Hendler and Golbeck 2008).

In this paper, we describe a particular scenario for using Linked Data: measuring semantic distance between resources to identify their relatedness. These measures could then be used in various applications, such as community detection in social networks, webpages suggestion for intelligent browsing and, as we focus in the second part of this paper, resources recommendations.

The rest of the paper is organized as follows. In the following section, we propose a theoretical definition of Linked Data and detail how semantic distance measures can be applied to it. In the second part, we present how such measures could be applied in the realm of recommender systems. Especially, we demonstrate how such recommendations can be multilingual and self-explanatory, simply by side-effect of the Linked Data principles, enabling intelligent agents exploiting this large amount of structured and interlinked data. Finally, we conclude the paper with an overview of some directions for future works on the topic.

Date	Datasets	Average % of growth / 30 days
2009-07-14	95	0.50
2009-03-17	93	10.00
2009-03-05	89	8.21
2008-09-18	43	1.58
2008-03-31	34	1.87
2008-02-28	32	1.10
2007-11-07	28	3.00
2007-10-08	25	2.44
2007-05-01	12	N/A

Table 1: Growing number of datasets in the LOD cloud

Measuring Semantic Distance on Linked Data

Theorizing Linked Data

Linked Data is generally introduced as structured data published and linked together following the four principles defined by (Berners-Lee 2006). However, while this definition makes sense from a pragmatic and programatic point of view, there is a need to ground it to a theoretical framework so that algorithms making use of it can be formally defined. Then, and considering that the goal of Linked Data (and of the Semantic Web, that we consider as a superset of the Web

*The work presented in this paper was funded in part by the Líon-2 project supported by Science Foundation Ireland under Grant No. SFI/08/CE/I1380.

[1] http://linkeddata.prg
[2] Based on (Bizer et al. 2007) and http://richard.cyganiak.de/2007/10/lod/
[3] http://esw.w3.org/topic/HCLSIG/LODD
[4] http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html

of Data) is to build a Giant Global Graph[5] of knowledge, we propose the following definition.

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 $i_i = \langle l_j, r_a, r_b \rangle$*

Scaling to the Web, the Linking Open Data cloud is then defined as the union of all the graphs G_i that are published (and interlinked) on the Web, i.e. $LOD = \bigcup_i G_i$.

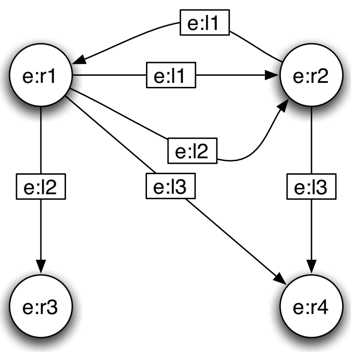

Figure 1: Example Linked Data graph

For example, the previous graph (Figure 1), in which we use the random namespace `http://example.org/` (prefix `e`), can be represented as:

$$\begin{aligned} R &= \{\texttt{e:r1}, \texttt{e:r2}, \texttt{e:r3}, \texttt{e:r4}\} \\ L &= \{\texttt{e:l1}, \texttt{e:l2}, \texttt{e:l3}\} \\ I &= \{\langle \texttt{e:l1}, \texttt{e:r1}, \texttt{e:r2} \rangle, \langle \texttt{e:l1}, \texttt{e:r2}, \texttt{e:r1} \rangle, \\ &\quad \langle \texttt{e:l2}, \texttt{e:r1}, \texttt{e:r2} \rangle, \langle \texttt{e:l2}, \texttt{e:r1}, \texttt{e:r3} \rangle, \\ &\quad \langle \texttt{e:l3}, \texttt{e:r1}, \texttt{e:r4} \rangle, \langle \texttt{e:l3}, \texttt{e:r2}, \texttt{e:r4} \rangle\} \end{aligned}$$

While the previous definition mainly focuses on resources and links (which are the core of the Linked Data principles), it is worth mentioning that ontologies are not excluded from that definition and are also taken into account in the proposed model. For instance, the `rdfs:subClassOf` link between `sioc:Post` and `sioc:Item` in the SIOC Core Ontology[6] can be modeled as $l = \langle \texttt{rdfs:subClassOf}, \texttt{sioc:Post}, \texttt{sioc:Item} \rangle$

Semantic Distance Using Linked Data

Based on the previous graph definition, we defined different strategies to compute semantic distance (Rada et al. 1989) (Budanitsky and Hirst 2001) between Linked Data resources. Semantic distance can be seen as a way to compute the relatedness between two resources (in our case, defined by their own URI). It is often used in ontology matching (Euzenat and Shvaiko 2007) but can have several other usages, such as identifying people sharing common interest to mine communities in social networks. While some approaches concentrate on the taxonomy of classes in the underlying ontologies (Rada et al. 1989), our focus is (1) to rely on any kind of link (and not only hierarchical ones), and (2) to rely only on links to compute these distances[7]. We defined a set of measures for semantic distance, named $LDSD$ (*Linked Data Semantic Distance*), with several variants that we will now describe. Each of them is tailored to provide results in the $[0, 1]$ interval, the smallest distance implying the most important similarity between resources[8].

Direct Distance Our first measure for semantic distance strictly relies on direct links between resources, both incoming and outcoming. Before going through the algorithm, let us provide the following definition.

Definition 2. *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, 0 if not. By extension C_d can be used to compute (1) the total number of direct and distinct links from r_a to r_b ($C_d(n, r_a, r_b)$) as well as (2) the total number of distinct instances of the link l_i from r_a to any node ($C_d(l_i, r_a, n)$).*

For example, in the example graph (Figure 1), we have:

$$\begin{aligned} C_d(\texttt{e:l1}, \texttt{e:r1}, \texttt{e:r2}) &= 1 \\ C_d(n, \texttt{e:r1}, \texttt{e:r2}) &= 2 \\ C_d(n, \texttt{e:r2}, \texttt{e:r1}) &= 1 \\ C_d(\texttt{e:l1}, \texttt{e:r1}, n) &= 1 \\ C_d(\texttt{e:l2}, \texttt{e:r1}, n) &= 2 \end{aligned}$$

Based on the previous definition, we defined a first similarity measure, named $LDSD_d$ (Figure 2), that simply considers the direct — incoming and outcoming — links between resources.

$$LDSD_d(r_a, r_b) = \frac{1}{1 + C_d(n, r_a, r_b) + C_d(n, r_b, r_a)}$$

Figure 2: Direct distance

For example, in the example graph, we have:

$$LDSD_d(\texttt{e:r1}, \texttt{e:r2}) = \frac{1}{1 + 2 + 1} = 0.25$$

We also provided a weighted version of that measure (Zhong et al. 2002), $LDSD_{dw}$ (Figure 3), in which the

[5] `http://dig.csail.mit.edu/breadcrumbs/node/215`

[6] `http://rdfs.org/sioc/ns`

[7] For example, we did not consider additional measures such as computing distance between the `dc:description` or `rdfs:label` values of the resources.

[8] While some of the following measures are not symmetric, we kept the term distance for all of them.

$$LDSD_{dw}(r_a, r_b) = \frac{1}{1 + \sum_i \frac{C_d(l_i, r_a, r_b)}{1+log(C_d(l_i, r_a, n))} + \sum_i \frac{C_d(l_i, r_b, r_a)}{1+log(C_d(l_i, r_b, n))}}$$

Figure 3: Direct distance, weighted

weight depends on the number of times each link appears in the graph — from the same resource to any other one — in order to gives less impact to the most popular links.

Indirect Distance Going further, and since the value of Linked Data resides not only in direct links between resources but also in shared connections through other resources, we designed a second version of the $LDSD$ algorithm, based on indirect links between resources. As previously, we shall introduce the following definitions before detailing our algorithms.

Definition 3. *C_{io} an C_{ii} are functions that compute the number of* indirect and distinct *links, both* outcoming and incoming*, between resources in a graph G. $C_{io}(l_i, r_a, r_b)$ equals 1 if there is a resource n that satisfy both $\langle l_i, r_a, n\rangle$ and $\langle l_i, r_b, n\rangle$, 0 if not. $C_{ii}(l_i, r_a, r_b)$ equals 1 if there is a resource n that satisfy both $\langle l_i, n, r_a\rangle$ and $\langle l_i, n, r_b\rangle$, 0 if not. By extension C_{io} and C_{ii} can be used to compute (1) the total number of indirect and distinct links between r_a and r_b ($C_{io}(n, r_a, r_b)$ and $C_{ii}(n, r_a, r_b)$), respectively outcoming and incoming) as well as (2) the total number of resources n linked indirectly to r_a via l_i ($C_{io}(l_i, r_a, n)$ and $C_{ii}(l_i, r_a, n)$), respectively outcoming and incoming)*

Based on this distance, in our example graph we have $C_{io}(\texttt{e:l3}, \texttt{e:r1}, \texttt{e:r2}) = 1$ (via outcoming links to `e:r4`) and $C_{ii}(\texttt{e:l2}, \texttt{e:r2}, \texttt{e:r3}) = 1$ (via incomings links from `e:r1`).

We then defined a first similarity measure taking these links into account, named $LDSD_i$ (Figure 4), and we also developed a weighted version, $LDSD_{iw}$ (Figure 5). As previously, this weighted version gives more weight to the less popular links, considering that two resources are more related if they are the only ones sharing a particular property (Passant et al. 2008).

$$LDSD_i(r_a, r_b) = \frac{1}{1 + C_{io}(n, r_a, r_b) + C_{ii}(n, r_a, r_b)}$$

Figure 4: Indirect distance

In our example, we then have:

$$LDSD_i(\texttt{e:r1}, \texttt{e:r2}) = \frac{1}{1+2+0} = 0.5$$

Combined Distance Finally, our latest measure combines the two previous ones, taking into account both the direct and indirect relationships that happen in a graph. As previously, we defined a simple version ($LDSD_c$) and a weighted one ($LDSD_{cw}$), the latter one being defined in Figure 6.

Use-case: Resources Recommendations

The similarity measures we presented so far can be used in various contexts, from identifying communities in social networks based on shared characteristics of their members (*e.g.* common interests) to intelligent interfaces suggesting relevant pages when browsing the Web. In order to demonstrate our findings regarding the previous algorithms, we developed a third use case: recommendation system. Our idea was to use these distances to build systems providing recommendations such as *"If you like X, you should like Y"*, because of the particular (direct and indirect) links that exist between two resources *X* and *Y*.

While the use of Semantic Web technologies has already been discussed in the context of recommender systems, for instance by (Middleton, Alani, and Roure 2002), (Celma and Serra 2008) and (Passant and Raimond 2008), our motivations were once again to see what could be achieved with the current state of the LOD cloud, simply relying on the measures expressed before and on the links available on the Web of Data. In addition, we will see how such recommendations can be self-explanatory, by side effect of the $LDSD$ algorithms combined with the Linked Data principles.

Ranking the Various Measures

As we designed six different versions of the $LDSD$ algorithm, our first goal was to identify which one would be the most suitable for such recommendations. To do so, we asked five users to submit a list of five to ten artists and bands, leading to a total of 39 distinct resources, with genres ranging from Country to New Wave, and we applied the different algorithms for resource, each one leading to a list of 10 recommended resources. Then, we asked the users to identify the best algorithm for each recommendation, the results being displayed in the following table (Table 2). In average, and in spite of some issues that we will discuss later, the $LSDS_{cw}$ algorithm was considered as the best one.

Algorithm	Times ranked first
$LDSD_d$	10
$LDSD_{dw}$	1
$LDSD_o$	2
$LDSD_{ow}$	8
$LDSD_c$	5
$LDSD_{cw}$	**13**

Table 2: User evaluation for $LDSD$ variants

Consequently, we concentrated on the $LDSD_{cw}$ variant and we will now describe how we applied it to build the aforementioned recommender system (we will simply use $LDSD$ to refer to $LDSD_{cw}$ in the rest of the paper).

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

Figure 5: Indirect distance, weighted

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

Figure 6: Combined distance, weighted

Application to the Musical Domain

In order to apply the $LDSD$ algorithm to the musical domain, we decided to concentrate on the DBPedia dataset[9]. One limitation of this approach is that we cannot benefit from links from others datasets to DBpedia (that could be used in the context of indirect links). Doing so would require to either concentrate on a replica of the whole LOD cloud (for instance via the endpoint provided at `http://lod.openlinksw.com/sparql`), use an index as Sindice[10] or study further how to find the right balance between distributed architectures and centralized systems for Semantic Web applications, as discussed in (Heitmann et al.). However, DBPedia already offers a large database for such resources recommendations, as it contains more that 39,000 distinct instances of `dbpedia-owl:MusicalArtist` or `dbpedia-owl:Band` for which recommendations could be provided.

Practically, our algorithm simply takes a seed URI as input to compute the distance between this URI and all other resources from the dataset. The distance is computed with the $LDSD$ algorithm that is translated in a set of SPARQL queries and processing instructions. To be more relevant, we limited the recommended resources to instances of either `dbpedia-owl:MusicArtist` and `dbpedia-owl:Band` (to avoid getting recommended a city, for instance) which also require some tuning when considering the weights of links. The next table shows the result of the $LDSD$ algorithm for a query about Johnny Cash[11], the recommended resources being ordered by semantic distance from the original resource.

While a future step, currently in progress, is to evaluate such recommendations based on user feedback — which was in a way already done by comparing the 6 different algorithms — and by comparing them to an existing baseline, a first analysis of the various recommendations lead to interesting results. For example, by considering the first 15 recommendations for Johnny Cash on Last.fm[12], six of them are included in the $LDSD$ results (actually, five of our first ten recommendations overlap with the Last.fm top 20 results). In addition, the most interesting part, in our opinion, is that the algorithm was able to identify bands that were not in the Last.fm suggestions, but are accurate recommendations. For example, The Tennessee Three, the backing band of Johnny Cash, appear in our results but not in the Last.fm recommendations[13] and is however a relevant recommendation for someone interested in that particular artist.

Yet, we identified several issues with the current algorithm. For instance, Justin Timberlake[14] was suggested as a relevant resource for Elvis Presley[15], as both have various geolocation properties in common. Giving more importance to a particular kind of links, in addition to the existing weights, might be a way to improve the algorithm to such extent and such could by done by considering (and mining) user interests (Debnath, Ganguly, and Mitra 2008) and using it in the weight definition so that the relatedness between two entities could also depend on some social factors and not only on statistical ones.

Artist	Distance
June Carter Cash	0.12816871468
Kris Kristofferson	0.13405547938
Elvis Presley	0.13984301344
Glen Campbell	0.15416659420
Willie Nelson	0.16082144980
The Highwaymen	0.16660898801
The Tennessee Three	0.17378105488
Dolly Parton	0.17785793324
Jerry Lee Lewis	0.17926729848
Jack Clement	0.18533334157
Bob Dylan	0.19315185867
Louis Jordan	0.19511230363
Charlie Rich	0.19664759571
Carlene Carter	0.19677362050
Al Green	0.19969509951

Table 3: Recommendations for Johnny Cash, ordered by distance from the seed resource

[9] `http://dbpedia.org`
[10] `http://sindice.com`
[11] `http://dbpedia.org/resource/Johnny_Cash`
[12] `http://last.fm`
[13] At least not in their 50 first answers.
[14] `http://dbpedia.org/resource/Justin_Timberlake`
[15] `http://dbpedia.org/resource/Elvis_Presley`

Application to the Literature Domain

Since the semantic measures that we proposed are completely agnostic of the kind of resources that they deal with, we also stressed the algorithm for book recommendations, without applying any change to it. The following table (Table 4) shows the recommended instances of `dbpedia-owl:Book` for the book "Fight Club"[16]. While the results are accurate (all are books written by the same author), their number is quite limited, because there are no much incoming and out-coming links to and from the original resource. Extending our algorithm to a recursive one such as SimRank (Jeh and Widom 2002) may then be useful where there are only a few links from and to the seed resource, as in that use-case.

Book title	Distance
Invisible Monsters	0.2275
Survivor	0.2290
Choke	0.2734
Diary	0.2880
Lullaby	0.2880

Table 4: Recommendations for the book "Fight Club", ordered by distance from the seed resource

User Interface: Multi-lingual and Self-explanatory

As the system provide results as resources URIs, these resources following the Linked Data principles and then being dereferencable and delivering RDF information about themselves, one advantage is that they all contain valuable informations that can be used to build the related user interfaces: description of the artist (`dc:description`), picture (`foaf:img`), etc. Moreover, some of these information are available in multiple languages. Then, we can benefit from it to build both (1) user-friendly user-interfaces, displaying not only the name of the recommended resources but also their picture and description and (2) provide multilingual capabilities for such interfaces, these two benefits coming for free as a side effect of the use of Linked Data to compute the distance, and consequently the recommendations. For instance, Figure 8 demonstrates a simple user-interface to browse the results of the previous recommendations for Johnny Cash in English (left) and Spanish (right), having pictures of the recommended artists in both cases, retrieved as well from DBpedia. These two interfaces are based on SPARQL queries and the only difference is the use of a difference value for the `FILTER` by `LANG` clause. We must however mention that this multilingual aspect can not always be achieved, as some resources descriptions may not be available in the requested language but only in english.

Finally, another important aspect, such system provides a way to explain the recommendations to the users, a trend that is generally seen as a way to make recommendations more transparent and more acceptable (McSherry 2005). For instance, considering the previous Johnny Cash example, as some recommendations can be seen as *niche* recommendations, they might not make sense for someone that has not a clear overview of the seed artist. Then, by explaining the recommendations, we are able to make them more interesting and better accepted by the users.

As $LDSD$ takes advantage of the links that exist between resources to compute their relatedness, we benefited from these links to explain the recommendations. Indeed, the algorithm keeps track of all the links (both direct and indirect) that have been used, making this information available to the end-user in a user-friendly way.

For instance, in the following picture (Figure 7), one can see that Big Brother and the Holding Company[17] has been recommended for a query regarding Janis Joplin[18] as, among others, they are associated musical artists (direct link) and both considered as people associated with the hippie movement as well as they play the same music genre (indirect links). We benefit from the `rdfs:label` of each link to provide such interface, as well as using the weights to display the number of other artists in the dataset sharing the same links, going further in the explanations.

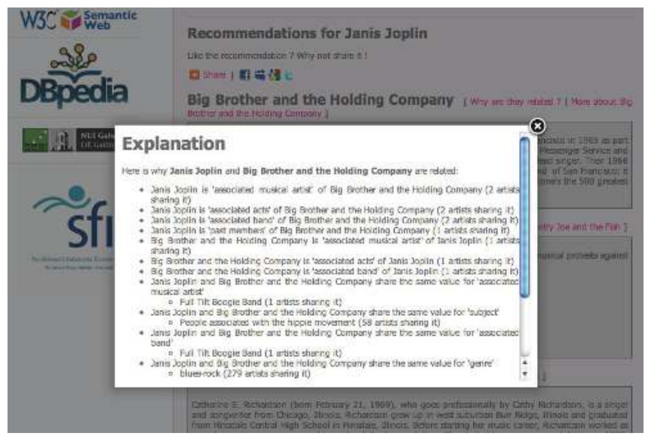

Figure 7: Explaining the recommendations

Conclusion and Future Works

In this paper, we detailed an approach to compute semantic distance on Linked Data, considering only the various links that can exist between Linked Data resources, and using both direct and indirect links. We discussed several *Linked Data Semantic Distance* algorithms ($LDSD$) that can be applied to measure relatedness between resources from the Web of Data and how they can be applied to provide resources recommendations. Especially, we demonstrated how these measures can be used to build a new kind of intelligent recommender systems, that are able to explain their recommendations to end-users, bringing the power of intelligent agents into the Web of Data.

[16] http://dbpedia.org/resource/Fight_Club

[17] http://dbpedia.org/resource/Big_Brother_and_the_Holding_Company

[18] http://dbpedia.org/resource/Janis_Joplin

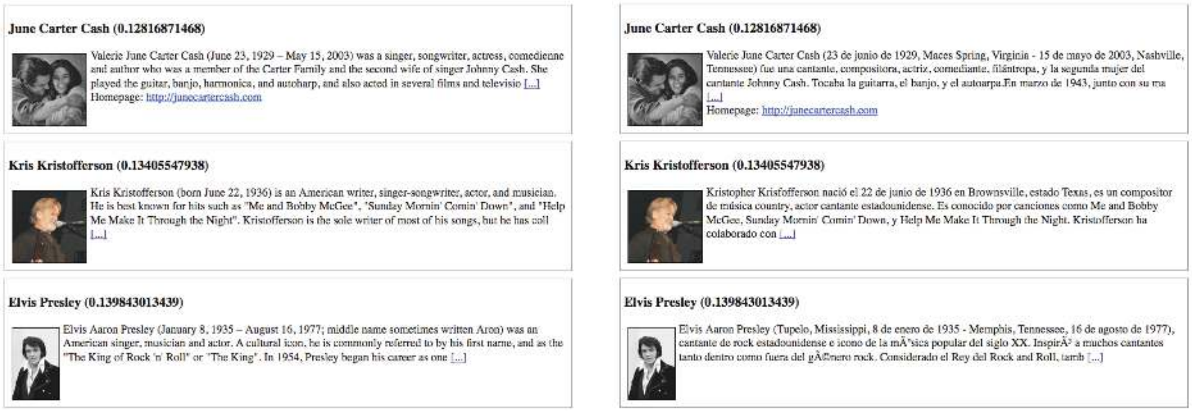

Figure 8: Multilingual interface for recommendations (English and Spanish)

In the future, different enrichments for the algorithms could be envisioned, such as adding new weight factors for the links used when computing the distances, using recursive measures as proposed by (Jeh and Widom 2002) or taking into account the links hierarchy, which could be useful especially when dealing with SKOS[19] — SKOS Simple Knowledge Organization System — hierarchies as in DBpedia.

Finally, we recently worked on consolidating the current implementation of $LDSD$ and built a complete knowledge base of musical recommendations, providing a service now available at `http://dbrec.net`, delivering self-explanatory recommendations for more than 39,000 bands and artists (Figure 7), also available as Linked Data.

[19] `http://www.w3.org/2004/02/skos/`

References

Berners-Lee, T. 2006. Linked Data. Design issues for the world wide web, World Wide Web Consortium. `http://www.w3.org/DesignIssues/LinkedData.html`.

Bizer, C.; Heath, T.; Ayers, D.; and Raimond, Y. 2007. Interlinking Open Data on the Web. In *Poster proceedings of the 4th European Semantic Web Conference (ESWC2007)*.

Budanitsky, E., and Hirst, G. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the NAACL 2001 Workshop on WordNet and other lexical resources*.

Celma, O., and Serra, X. 2008. Foafing the music: Bridging the semantic gap in music recommendation. *Web Semantics: Science, Services and Agents on the World Wide Web* 6(4):250–256.

Debnath, S.; Ganguly, N.; and Mitra, P. 2008. Feature weighting in content based recommendation system using social network analysis. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, 1041–1042. New York, NY, USA: ACM.

Euzenat, J., and Shvaiko, P. 2007. *Ontology Matching*. Berlin-Heidelberg: Springer-Verlag.

Heath, T. 2008. How Will We Interact with the Web of Data? *IEEE Internet Computing* 12(5):88–91.

Heitmann, B.; Kinsella, S.; Hayes, C.; and Decker, S. Implementing Semantic Web applications: reference architecture and challenges. In *Proceedings of the 5th Workshop on Semantic Web Enabled Software Engineering*, volume 524 of *CEUR Workshop Proceedings*. CEUR-ws.org.

Hendler, J. A., and Golbeck, J. 2008. Metcalfe's law, Web 2.0, and the Semantic Web. *Journal of Web Semantics* 6(1):14–20.

Jeh, G., and Widom, J. 2002. Simrank: a measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 538–543. New York, NY, USA: ACM.

McSherry, D. 2005. Explanation in Recommender Systems. *Artificial Intelligence Review* 24(2):179–197.

Middleton, S. E.; Alani, H.; and Roure, D. D. 2002. Exploiting synergy between ontologies and recommender systems. *CoRR* cs.LG/0204012.

Passant, A., and Raimond, Y. 2008. Combining Social Music and Semantic Web for music-related recommender systems. In *Proceedings of the ISWC2008 Workshop on Social Data on the Web*, volume 405 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Passant, A.; Mulvany, I.; Mika, P.; Maisonneauve, N.; Löser, A.; Cattuto, C.; Bizer, C.; Bauckhage, C.; and Alani, H. 2008. Mining for social serendipity. In *Dagstuhl Seminar on Social Web Communities*.

Rada, R.; Mili, H.; Bicknell, E.; and Blettner, M. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics* 19(1):17–30.

Zhong, J.; Zhu, H.; Li, J.; and Yu, Y. 2002. Conceptual graph matching for semantic search. In Priss, U.; Corbett, D.; and Angelova, G., eds., *Proceedings of the 10th International Conference on Conceptual Structures (ICCS 2002)*, volume 2393 of *LNCS*, 92–196. Springer.