International Journal of Advance Innovations, Thoughts & Ideas

Kushwaha et al., Int J Adv Innovat Thoughts Ideas 2013, 2:4
DOI: 10.4172/2277-1891.1000148

Review Article Open Access

Utilizing DBpedia via CBR Approach of Recommender System

Nidhi Kushwaha*, Rajesh Mahule and Vyas OP

Indian Institute of Information Technology, Allahabad, India

Abstract

Using Linked Open Data Cloud for knowledge extraction is a challenging as well as budding research field. Knowledge extraction is essential work of developing a Recommendation System (RS). Traditional context analyzers of Content Based Recommendation (CBR) are no more sufficient in current web era of semantics. This problem can be removed through generation of features from the logic of semantics presents implicitly in structured format in the web. This promising area merge various domains and technologies namely, semantic web, machine learning, personalization and information retrieval for achieving a good recommendation.

This paper discussed about the research questions and challenges that originate from the extraction of semantic knowledge. This semantics can be accessed from a huge open source data cloud that are linked in meaningful way and named as Linked Open Data Cloud (LOD). The implementation includes key methods for how to gain semantically enriched data related to particular items. Paper explains the book domain residing inside the DBpedia datasets, which consist of cross domain information.

Keywords: Semantic web; Linked data; Recommendation system; Content based

Importance and Need of LOD

Nowadays, the Internet enters in the stage where the information is being pushed by the people, for the people without considering their format. A revolutionary change in this continuation has been explored by the Linked Open Data Cloud Project [1]. Web contain heap of information, to gain knowledge from it, there is a need of technologies for handling the silos of data and provide only useful nuggets. It instigates the need of Recommender System, which means to provide on-time informative knowledge. Before the semantics, Recommender Systems are totally dependent on the knowledge of unstructured, semi-structured web or HTML (more specifically). These unstructured formats contain information that need suitable preprocessing for gaining content this is known as "content analysis" in Recommender System. Linked open Cloud comes into web since the beginning of 2008 by Sir Tim Berner's Lee [1,2]. It takes the web into new higher position where one can share their knowledge by publishing data in one format i.e. RDF (Resource Description Framework). One can use the knowledge by SPARQL [3] querying on the RDF [4], and more than that utilize it by means of different applications. RDF comes as the counterpart of the traditional approaches but in a new and more efficient way, of data format and availability. Linked Open Data cloud [2] conceives semantic web with exploitation of linkages between concepts in the form of Ontologies [5]. Ontology is not a new term but it will take long time to make sufficient progress for converting the whole web into the concepts and relationships form. Linked Open Cloud is the collection of different but dependent Ontologies. These Ontologies is associated by manual or automatic tools [6]. Worldwide approval of LOD can be seen by the emergence of its dependent applications. Many research works [7] are going on for predicting the links between different datasets inside the Linked Open Data Cloud for the efficient utilization of this openly available web data. The pivot of the LOD is form by a giant encyclopedia termed as DBpedia [8,9] (the RDF version of Wikipedia info box) from which the other domain specific datasets are connected in a complementary way. There is a need to manage this huge amount of knowledge that present in both unstructured and structured format. Confinement of different domains in LOD is consisting of information of diverse sources like music, movie, drugs, geographic, government, protein sports, news and much more different information [8]. The information is in the form of triples that means subject, predicate, and object. LOD are based on five principles [1], one of which is to provide URIs to each and every resource for give it a universal uniqueness. This uniqueness of the items are helpful for gaining the information related to them throughout cross domain datasets like DBpedia, Freebase and also specific domain datasets like LinkedMDB[1] for movie, DBTune[2] for music. In the next section authors are explained related work in the field of LOD recommendation.

Related Work

Fetching information from a Knowledge Base which is not complete is of no use. Although datasets that consist in the LOD cloud not lie on this category, but they have inconsistencies in terms of their relationship. For example: In DBpedia book domain some important properties like "publisher", "genre" is missing from some books. To gain the knowledge form the datasets it should be complete in all sense but in the other side achieving this is a hard problem for Miner. Previous approaches FeGeLOD [10], ExpLOD [11] uses the data sets that are not complete and enhance it with the help of DBpedia and YAGO ontology [12] of LOD Cloud. This enhancement depends on the hardcoded SPARQL queries embedded in the code. LOD Endpoints also gives different results from each other's due to inconsistency of data storage. So, to utilize this knowledge is a challenging issue in sense of incomplete linkage facilities and inconsistencies. Mitigation of this problem is also a challenging task. This task includes generating more and more links between different datasets inside the LOD cloud.

***Corresponding author:** Nidhi Kushwaha, Indian Institute of Information Technology, Allahabad, India, E-mail: kushwaha.nidhi12@gmail.com

Received August 03, 2013; **Accepted** August 27, 2013; **Published** September 02, 2013

Citation: Kushwaha N, Rajesh M, Vyas OP (2013) Utilizing DBpedia via CBR Approach of Recommender System. Int J Adv Innovat Thoughts Ideas 2: 148. doi: 10.4172/2277-1891.1000148

Although the RDF uses semantics to represent the data, it still have visualization problem. Various tools [13,14] are proposed for this purpose specifically. They all are based on nodes & links visualization. Visualization makes easy for quick understanding of the whole RDF graph. After the understanding of RDF graph we can apply various techniques to mine it or to gain information from it. Authors [14-17] proposed similarity matching and ML techniques like SVM to measure the similarity between two items and ultimately the results in RS. Approach [17] are considering only two hop depth for calculating weightage of a particular property we are here considering more than two hop. We are here discussing the key points and our finding while moving forward with the different approach of weight calculation for predicates related to the items (book domain).

Proposed Work

Authors are proposing a framework for weightage calculation form the given linkage properties of LOD. The knowledge base has linkage power, through which we can calculate which property or link is more suitable or appropriate for given item or seed point. The importance of the properties is dealing with the various attributes discussed with the help of definitions of the different links. Triplet form consists of <subject, predicate, object>. These objects act as a data source for particular a subject. Linkage between one subject to another object provides the capability of moving automatically form one source to another semantically. Here we describe about the linkage property which play significant role for particular item. These properties are defined with consideration of particular domain where the seed point is items which can be Movie, Music, Drug, Book or any Product.

Definitions

InComg: Graph 'G', is form using collection of unique Subject, predicate, Object i.e <$S_iP_iO_i$>. Here we consider that Graph 'G' denotes the datasets related to the particular domain. Let us assume a Subject 'S' as a seed point then all the links that act as object 'O' for 'S' is comes under "InComg" links. To count whole "InComg" gives us an idea of bag of resources that are connected to the particular seed point or Item 'S'.

OutGog: In 'G'= <S_i,P_i,O_i>, if we consider object 'O' as our interest then all the linkage that points to it, is "OutGog" links. To count whole "OutGog" gives us an idea of bag of subjects or Items that are connected with objects.

ItmLinPs: In 'G'= <S_i,P_i,O_i>, if we consider specific property 'Pi' as our interest with the specific Item or subject 'Si' then all connected links are "ItmLinPs".

PLinItm: In 'G'=< S_i,P_i,O_i>,'PLinItm' can be calculated using counting the particular Item 'Si' related to the object 'Oi' with the specific property 'Pi'. It illustrate that how much a particular property is powerful to explain an Item or subject.

PLinAll: In 'G'=<S_i,P_i,O_i>, we can calculate all the subjects that are related to different Items with the specific property and named as PLinAll. It is similar to the above property here we not consider only one specific Item.

ItmLinSamAs: LOD dataset "DBpedia" contains one very important predicate known as OwlSameAs. It has information about all the other terms in the Graph that are related to the particular Item or subject 'S_i'.

ItmDcSkosLin: Another more important property is Dcterms: subject that explains the resource or 'O_i' in more general terms. Skos: broader property describes categories related to Items particular object. Count of Dcterms: subject and Skos: broader gives us deeper understanding of the particular Item's object. We are explaining the above definitions with the help of SPARQL query in Book domain in the Table1.

For Calculation of weightage of property we believe on a formula which is explained below. This will calculate the importance on a particular property in the whole RDF graph (Book graph in this case). Here we consider all the links that have the specific property (considering in the whole domain specific graph) as a denominator and number of link that one specific item consist related to that property. Like for example a specific Harry Potter book has total 42 links related to author property and total number of author property links are 38159 in the whole Book domain graph. Then the weightage of particular property is 2.88 for a specific Harry Potter book. Overall weightage is calculated by multiplying it with the total property links that are connected with specific Harry Potter book (including all properties author, publisher and genre) divided by total number of books in the whole domain specific RDF graph. By this way we can calculate the importance of particular property among all the properties for a specific book.

Formulae Used:

1. Weight for particular Property "α_p"

α_p= PLinItm/ PLinAll

PLinItm and PLinAll are explained in above.

2. Overall weight calculation>

Overall_weight = $\alpha p * \mathbf{ItmLinPs}/1+\log T$

Here, T is the total number of items or subject that we consider for particular domain. By applying above formulae we obtained following matrix, where rows are populated with Items or subject s and columns

Different Links	*SPARQL Query*
InComg	SELECT distinct count(?in) {?s a dbpedia-owl:Book. ?o ?in ?s.}
OutGog	SELECT distinct count(?out) {?s a dbpedia-owl:Book.?s ?out ?o.}
ItmLinPs	SELECT distinct count(?Ps) {< http://dbpedia.org/resource/The_Hollow> ?Ps ?o.}
PLinItm	
AutLinItm	SELECT distinct count(?author) {?s a dbpedia-owl:Book; dbprop:author ?author.}
GenreLinItm	SELECT distinct count(?type) {?s a dbpedia-owl:Book. ?s dbprop:genre ?type.}
PubLinItm	SELECT distinct count(?publisher) {?s a dbpedia-owl:Book; dbprop:publisher ?publisher.}
PLinAll	SELECT distinct count(?p) {?s a dbpedia-owl:Book; ?p ?o.}
ItmLinSameAs	SELECT distinct count(?o) {?s a dbpedia-owl:Book ?s owl:sameAs ?o.}
ItmDcSkosLin	SELECT distinct count(?dcterm) count(?o) {?s a dbpedia-owl:Book ; dcterms:subject ?dcterm. ?dcterm skos:broader ?o}

Table 1: Definition description.

are populated with properties of the Items fetched from DBpedia. The matrix is filled by the weights. After the application of above formula we get values of Table 2 Next section gives brief description of the implementation details.

Implementation

Identifying relevant datasets from dbpedia

DBpedia is the RDF encyclopedia, from which we have to extract selected information for further computations. It consists of 103 million triples incorporation of different information related to Animals (1,91,536), Species (2,69,060), Mammals (17,902), Persons (7,52,613), Cities (17,27,060), Places (12,48,585), organizations (2,11,525), Historic Places (5735), Historic Buildings (3435) , Drug (4,854), Companies (52,562), Films (1,23,530), Songs (5441), Albums (1,77,056), Museums (3023), Books (1,07919), Written Works (1,20,712) triples with total size 3GB. Querying through the endpoint [18] via internet takes more time then the querying with local repository. Today various RDF triple stores (Sesame [19], Allegrograph [20], OpenLink Virtuoso [21], Fuseki [22], 4Store [23], StarDog [24]) are present that have the capability to store the RDF triples in efficient manner for quick consumption. For testing purpose we choose Sesame 2.7, that can work with windows system with 3GB RAM. We store approx 50,595 triples of Book domain. With the help of Sesame Java API we are able to perform query on the Sesame triple store. We found various inconsistencies in the DBpedia Book dataset. Like the DBpedia illustrator property has 2478 and 1859 links (in the whole book graph) with the http://dbpedia.org/property/ and http://dbpedia.org/ontology/ respectively. Here in Table 3 we have mentioned some important properties that have larger links then others.

Thanks to the property path facility of SPARQL Query Language [3] that provide traversal path extraction through one to more distance from the seed point. This facility comes with SPARQL 1.1, the SPARQL query evaluation. To minimize the burden on the Sesame triple store we fetched subject's category classification hierarchy (SKOS ontology) from the Virtuoso SPARQL endpoint. Deep extraction (more than 2 hop) of the hierarchy gives more generalized results in case of the skos: broader property which are not used in the previous approaches. By experiment we come to know that three levels deeper gives much generalized results like for category "Fantasy Anthologies Series" it give "Fantasy Anthologies" at level 1(1L), "Fiction Anthologies" at level 2 (2L) and which is further related to the category "Literature By Genre" at level 3 (3L) as mentioned in the Figure 1.

Book Name	Genere	Author	Publisher	Dcterms:subject	Owl:sameAs
The_Murder_of_Roger_Ackroyd	0.5523	0.0913	0.0294	0.019	0.0156
The_Mystery_of_the_Blue_Train	0.3474	0.043	0.0185	0.0119	0.0098
Elephants_Can_Remember	0.2663	0.044	0.0283	0.00367	0.0075
Life,_the_Universe_and_Everything	0.9162	0.0454	0.039	0.0151	0.0104

Table 2: Weight of different properties.

Property	Links
type	240478
subject	9965
SameAs	48440
author	26634 (property)
author	26457 (ontology)
publisher	25015 (property)
isbn	21132 (ontology)
isbn	2085 (property)
mediaType	20393 (property)
rdflabel	20875
illustrator	1859 (ontology)
illustrator	2478 (property)
previous Work	7647 (ontology)

Table 3: Property links description (only some).

Although more generalized term "Literature By Genre" at level-3 are less informative, but we have also taken this into consideration of calculating the weights of the properties. After calculating weights regarding each book, the similarity metrics can be calculated. Table 4 shows the relatedness of the book URI "http://dbpedia.org/resource/The_Murder_of_Roger_Ackroyd" with the other books inside the DBpedia using Pearson Correlation similarity metric. Result of the semantic measures are describes with the following Table 4. In the table it is easily determine that the books sharing same genre, authors, and publisher have great number of similarity rather than others. For example first recommendation of the book for "The Murder of Roger Ackroyd" have same author "Agatha Christie" and they both shared same genre "Crime Novel". So, if a person like the Crime Novels then its related books would be their right choice in future. This type of recommendation is called Content Based Recommendation (CBR). Web 3.0 gives the opportunity to the researcher to directly choose the related terms (objects) of the items (subject) from the statements (NTriples) considering their linkage type (predicates) like genre, publisher, and illustrator. Items sharing same objects in the RDF graph will be similar to each other because have something common. For Example, we know the fact that most people are interested to read the sequel of books, written by the same author. In the next section we discussed shortcomings and future scope related to our approach.

Discussions and Future Scope

LOD is act as an evaluation of the knowledge base that contains huge interlinked information to share. In our dataset we consider only DBpedia dataset, while in future we definitely work with more than two or three related RDF datasets. Domain analysis also raises the quality of application specific tools because it consist specific information from different sources. The Giant Cloud has millions of triples which can be mine to gain interesting things. Recommending the predicates by ARM (Association Rule Mining) [25] can be possible if we consider the name of Books as the Context and mine the predicates (Authors, Publisher, Writer). While on the other hand mining the Book names in the context of their objects (J.K. Rowling, Fantasy book, Harry_Potter Series) cluster the similar types of books on the basis of their frequently co-occurring. In future we mine with the different datasets in various contexts to gain recommendation from the triples. We will apply this knowledge to cluster the same type of books by matching their user specific properties. Like if a person give more interested to read a book of particular author or particular publication we can able to recommend him/her without worrying other properties of the book.

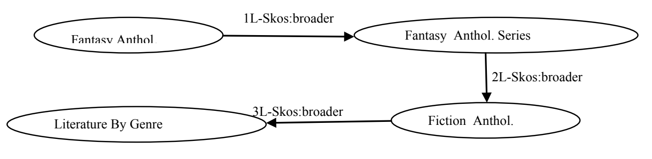

Figure 1: Levels description.

Book	Relatedness
http://dbpedia.org/resource/Death_in_the_Clouds	1.745754330
http://dbpedia.org/resource/International_Encyclopedia_of_Human_Geography	1.637414408
http://dbpedia.org/resource/Elephants_Can_Remember	1.625657835
http://dbpedia.org/resource/Falling_%28Provoost_novel%29	1.605444268
http://dbpedia.org/resource/Murder_in_Mesopotamia	1.582832742
http://dbpedia.org/resource/The_Mystery_of_the_Blue_Train	1.548746678
http://dbpedia.org/resource/I_Have_Landed	1.446938096
http://dbpedia.org/resource/Ice_and_Fire	1.438493147
http://dbpedia.org/resource/Life,_the_Universe_and_Everything	1.409638694

Table 4: Shows the relatedness of the “http://dbpedia.org/resource/The_Murder_of_Roger_Ackroyd” with others.

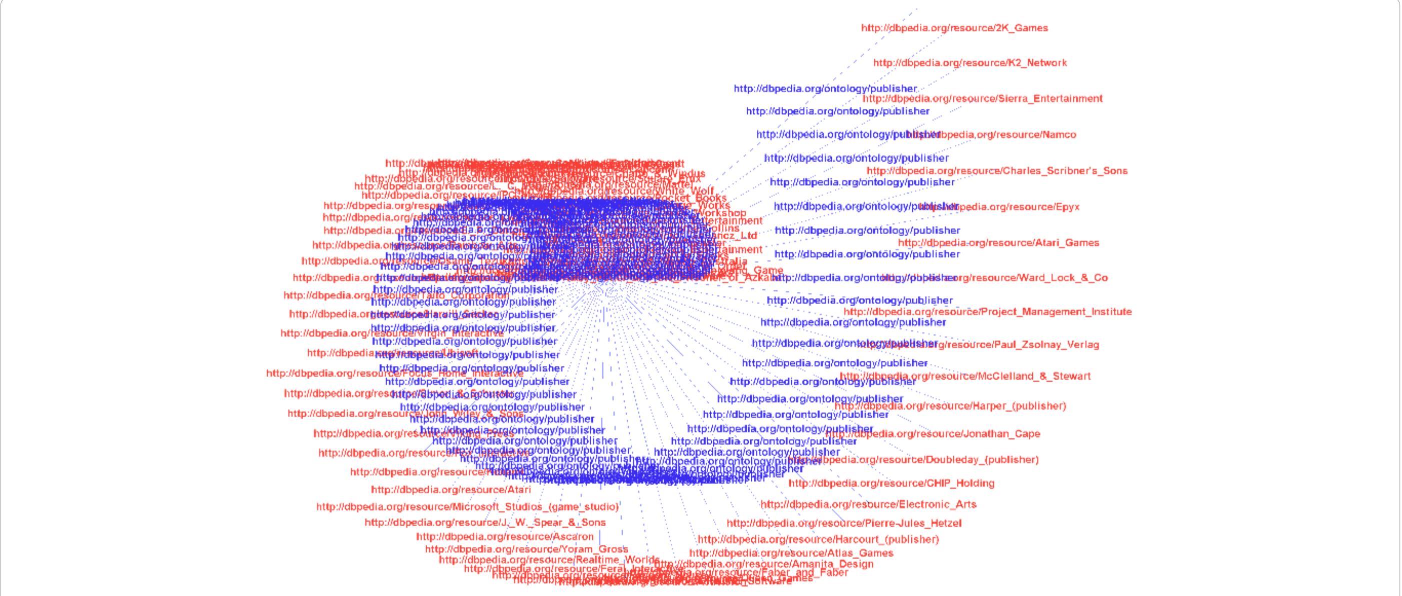
http://dbpedia.org/resource/2K_Games
http://dbpedia.org/resource/K2_Network
http://dbpedia.org/ontology/publisher
http://dbpedia.org/resource/Sierra_Entertainment
http://dbpedia.org/resource/Namco
http://dbpedia.org/resource/Charles_Scribner's_Sons
http://dbpedia.org/resource/Epyx
http://dbpedia.org/resource/Atari_Games
http://dbpedia.org/resource/Ward_Lock_&_Co
http://dbpedia.org/resource/Project_Management_Institute
http://dbpedia.org/resource/Paul_Zsolnay_Verlag
http://dbpedia.org/resource/McClelland_&_Stewart
http://dbpedia.org/resource/Harper_(publisher)
http://dbpedia.org/resource/Jonathan_Cape
http://dbpedia.org/resource/Doubleday_(publisher)
http://dbpedia.org/resource/CHIP_Holding
http://dbpedia.org/resource/Electronic_Arts
http://dbpedia.org/resource/Pierre-Jules_Hetzel
http://dbpedia.org/resource/Harcourt_(publisher)
http://dbpedia.org/resource/Atlas_Games
http://dbpedia.org/resource/Amanita_Design
http://dbpedia.org/resource/Faber_and_Faber
http://dbpedia.org/resource/Atari
http://dbpedia.org/resource/Microsoft_Studios_(game_studio)
http://dbpedia.org/resource/J._W._Spear_&_Sons
http://dbpedia.org/resource/Ascaron
http://dbpedia.org/resource/Yoram_Gross
http://dbpedia.org/resource/Realtime_Worlds

Figure 2: Visualization of publisher property of the book “harry potter and the prisoner of Azkaban”.

Conclusion

In this paper we have discussed about the advantage of Web 3.0 in the field of Recommender System. Although the paper based on the suitability of the Content Method with Web 3.0, Collaborative Filtering is not totally fit for that. It is because of the lack of the reviewing information in the LOD Cloud. Reason for this can be better understand in terms of privacy preservation of personal data. Content based approach also has some difficulties in it. One of them is to extract knowledge from LOD Cloud. It is not easy due to unclear boundary of the graph. Visualizing the graph as a whole is also not the solution for it. We have used Semantic web plug-in of the Gephi [13] tool to visualize the only one (publisher) property of the Harry potter book describe in Figure 2. Visualization of RDF graph gives us a realization of how different properties are related to different subjects and ease of SPARQL querying on them. In the above approach we manually select the interesting features of the item like publisher, genre etc. In future we will try to apply automatic approach of selection of properties to give our system more automation. Presently we are only using DBpedia dataset that are very noisy and huge. For the future point of view we will integrate more domain specific RDF datasets to give reality to our system.

References

1. Bauer F, Kaltenböck M (2011) Linked Open Data:The Essentials. edition mono/monochrom, Vienna, Austria.
2. Bizer C, Heath T, Berners-Lee T (2009) Linked data - the story so far. Int J Semantic Web Inf Syst.
3. Prud'hommeaux E, Seaborne A (2008) Sparql query language for rdf. W3c recommendation. W3C.
4. Klyne G, Carroll JJ (2004) Resource description framework (RDF): Concepts and abstract syntax. Technical report, W3C.
5. Kiefer C, Bernstein A, Locher A (2008) Adding data mining support to sparql via statistical relational learning methods. In Proceedings of the 5th European semantic web conference on The semantic web: research and applications 478-492.
6. Maedche A, Staab S (2001) Ontology learning for the semantic web. IEEE Intelligent Systems 16: 72-79.
7. Nguyen K, Ichise R, Le B (2012) SLINT: A Schema Independent Linked Data Interlinking System. Proceedings of the 7th International Workshop on Ontology Matching 1-12.
8. Bastian M, Heymann S, Jacomy M (2009) Gephi: An Open Source Software for Exploring and Manipulating Networks. International AAAI Conference on Weblogs and Social Media.
9. Broekstra J, Kampman A, van Harmelen F (2012) Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema, ISWC.
10. Franz Inc (2010) AllegroGraph RDFStore Web 3.0's Database.
11. Erling O, Mikhailov I (2009) RDF support in the virtuoso DBMS. Stud Comput Intell 221: 7-24.
12. Harris S, Lamb N, Shadbolt N (2009) 4store: The Design and Implementation of a Clustered RDF Store.

13. Voigt M, Mitschick A, Schulz J (2012) Yet Another Triple Store Benchmark? Practical Experiences with Real-World Data. Proceedings of the 2nd International Workshop on Semantic Digital Archives.

14. Dbpedia data.

15. Mendes PN, Jakob M, Garcia-Silva A, Bizer C (2011) Dbpedia spot-light: shedding light on the web of documents. Proceedings of the 7th International Conference on Semantic Systems, New York, NY, USA.

16. Paulheim H (2012) Explain-a-lod: using linked open data for interpreting statistics. International Conference on Intelligent User Interfaces IUI.

17. Paulheim H, Furnkranz J (2012) Unsupervised generation of data mining features from linked open data. WIMS, ACM.

18. Suchanek FM, Kasneci G, Weikum G (2007) Yago: a core of semantic knowledge Unifying WordNet and Wikipedia. Proceedings of the 16th international conference on World Wide Web, New York, NY, USA.

19. SparqlEndpointDescription.

20. Varga B, Groza A (2011) Integrating DBpedia and SentiWordNet for a tourism recommender system. IEEE International Conference on Intelligent Computer Communication and Processing.

21. Passant A (2010) dbrec: music recommendations using DBpedia. In Proceedings of the 9th international semantic web conference on the semantic web.

22. Ostuni VC, Noia TD, Mirizzi R, Romito D, Sciascio ED (2012) Cinemappy: a context-aware mobile app for movie recommendations boosted by dbpedia.

23. Semantic Visualization with Gurff.

24. Abedjan Z, Naumann F (2013) Improving RDF Data Through Association Rule Mining, In Datenbank-Spektrum 13: 111-120.

25. Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2008) Statistical properties of community structure in large social and information networks. In: Proceedings of the 17th international conference on World Wide Web (WWW '08), ACM, New York, USA.