I J C T A, 9(28) 2016, pp. 235-241
© International Science Press

HANDLING BROKEN LINK WITH MODEL BASED SIMILAR LINKS SUGGESTION IN LINKED OPEN DATA (LOD)

K.Lokeshwaran* and A. Rajesh**

Abstract: The emergence of the Linked Open Data (LOD) gave birth to a variety of open knowledge bases that can be freely and easily accessible on the Web .The LOD initiative has stimulated various institutions and organizations to publish their data on the web in a meaningful way and to interlink them with those of other data sources. But LOD sources are not stable they often change, so the links between resources will become obsolete (i.e. broken links) and which leads to processing errors in applications that consume web of data. Ignoring this problem is not the best approach but unfortunately it is the current best possible solution around and leaving the burden to the applications which consume the linked data. But apart from the current approach what we think is, the LOD data sources should provide the high link coherence in order to alleviate applications from this freight. As a possible solution, instead of leaving burden to the consuming application, Model based similar links can be suggested when broken link is detected. Our approach adapts support vector machine (SVM) to deal with RDF triples. In this paper we have used DBpedia and librarything LOD datasets to suggest similar links using model based approach.

Keywords: Linked open data, DBpedia, SVM, RDF, Model Based RS.

1. INTRODUCTION

Web is a classic source where information's are available almost at a free will. That's why people generally call web as global data source. But the problem with classic web is, the information's are not structured in a proper way. Due to the unstructured nature of web it can't make sure that the information retrieved by the user's are semantically correct. The information's are retrieved based on the metadata, keywords etc., which is specified in the web document. With the advent of semantic web, it is possible to present web content in a structured and semantic way. Which will be useful for the user's to retrieve content in a meaningfully. Another important advantage of semantic web is that one resource will link to other semantically related resources via semantic links. Although the coupling between the semantic links will break [7], when the resources are removed or moved. This leads to processing errors in applications which consume semantic data. So, in order to handle broken link by suggesting similar links, in this paper we show how our model based[1][11] approach efficiently suggests similar links when broken link is found.

The rest of the paper is organized as follows: in section 1.1 and 1.2 we have given brief overview about linked open data and broken link problem. In section 2 we illustrate the related work in handling broken links in LOD[8]. In section 3 we elucidate our model based approach to suggest similar links. Section 4 is set apart for evaluation. Finally in section 5 we give a precise conclusion and future work to finish the paper.

1.1 Linked Open Data

In this modern world, much of the knowledge (data) we get from the web is handed over to us in the form of HTML documents. HTML documents are linked to each other with the help of hyperlinks.

* Research Scholar, Department of Computer Science & Engg SCSVMV University, Kanchipuram. **Email:** k.lokeshwaran@gmail.com
** Professor and Head, Department Computer Science & Engg C. Abdul Hakeem College of Engg & Tech Melvisharam, India. **Email:** Amrajesh73@gmail.com

Humans are capable of reading these HTML documents, but machines have trouble to extract any meaning from these HTML documents themselves. Nevertheless HTML documents are connected via hyperlinks, there should be a quality mechanism for specifying the existence and to provide meaning between the items described in the document. The development of generic standards enables the Web to evolve various technical architectures. Hyperlinks allow the users to navigate between different documents. It also enables the capabilities of search engines to crawl the Web and to provide complicated search performance on the top of crawled content. Hence Hyperlinks are the important in connecting content from different location into a single global information space. Linked Data intervene directly on Web architecture and utilize this architecture to the task of partake data on global space.

So, the basic idea of Linked open Data is to apply the general architecture of the World Wide Web to the task of partake structured data on global scale. This mechanism is provided by the Resource Description Framework (RDF). The RDF provides a flexible way to describe things in the world, such as people, locations, or abstract concepts and how they relate to other things. These statements of relationships between things are, in essence, links connecting things in the world. In a Linked Data context, the RDF link connects URIs in different namespaces; it ultimately connects resources in different data sets. A Linked Data application that has looked up a URI and retrieved RDF data describing a person may follow links from that data to data on different Web servers, describing, for instance, the place where the person lives or the company for which the person works. So, the Web of Data is based on standards and a common data model, it becomes possible to implement generic applications that operate over the complete data space.

1.2. The Broken Link Problem

The content of Linked Data sources changes, so the data about new entities is added, outdated data is changed or removed. RDF links between data sources are updated at irregular intervals which lead to broken links pointing at URIs that are no longer maintained. These broken links will create more problems in LOD consuming applications therefore, we believe link integrity as a qualitative property that is given when all links within and between a set of data sources are valid and deliver the result data intended by the link creator. There are two ways in which links are broken they are *structurally* and *semantically* broken links.

Structurally Broken link: The link is considered as structurally broken if its target resource had representations that are not retrievable anymore.

Semantically Broken link: The link is considered as semantically broken if the human interpretation (the meaning) of the representations of its target resource differs from the one intended by the link author.

It is much harder to detect the semantically broken link than the structurally broken ones. In our approach we are not trying to identify the cause for broken link instead we are going to recommend similar links when broken link is found.

2. RELATED WORK

We have the strong notion that in future linked open data is going to rule the cyber space. But this will happen only when, the problems in LOD are eradicated. One such problem that threatens the growth of LOD is broken link. Usually links will become invalid when the indented resource is modified or relocated. The unavailability of resource from specific link will incur problems in LOD data consuming applications. So, it is mandatory to handle broken link. There are few quite impressive methods are there to handle broken link. Here we have discussed few approaches towards handling broken link. The simplest method to handle broken link is to simply *ignore the problem* and leave the burden to higher level application that acquires data. But this is not a highly recommended practice and not suitable for LOD.

The most common and preferable approach is *Embedded Links* representation. Usually classic web uses this mechanism by referencing the target resource in a source document via HTTP URI reference. This approach preserves high link cohesion. *Relative References is* relatively a soft approach to avoid broken links. But it works well only when entire resource collection is relocated

Indirection, a kind of translation mechanism which uses aliases for the links. Aliases point to the target resource location and require translation service to translate between an alias and its pointing location. Service translation tables should be updated whenever resources are relocated or deleted. Due to its centralized nature this approach is not suited for all environments.

In *Versioned and Static Collections*, the resources are archived and no modifications/deletions can be performed on this collection. So, the links those are all out of this collection are prone to obsolete.

Regularly updating the links whenever it is modified, this allows the application to redirect to the new resource location. Another approach is to keep *redundant* copies of resource. If a resource is not available on a particular link, a redirection service forwards referrers to another location where one of these copies exists. This approach is a combination of versioning and indirection approaches.

Based on the current state, the computations are performed to generate a *dynamic link*. But generating a dynamic link is not an easy task because it depends on lot of external factors, for example the dynamic link should be context dependent. Currently no one in the world is adopting this approach.

The *Detect and* Correct mechanism is the most preferable methodology. In this approach whenever the application using the link it first checks the validity of the resource endpoint reference against the centralized information which is already available, on the off chance that the legitimacy test comes up short, an endeavor to redress the connection by moving it.

Currently there is a tool called DSNotify[19] to handle the broken link using detect and correct mechanism. DSNotify periodically monitors items in a specific linked data source and extract descriptive feature vector for each item. Then it index the item +feature vector. Later it uses the feature to detect if items are removed or moved to another location. If the item is moved the new location, then the old and new relationship between the item is registered. The problem with this tool is periodic monitoring is required which is quite expensive and it is based on centralized mechanism.

Alternate Suggestion: The solution for handling the broken link problem proposed in this work falls mainly into this category. When a broken link is found instead of sending HTTP error response to the LOD consuming application an alternate similar links can be suggested. But the hectic lies in how the alternate links are suggested.

3. MBSLS-MODEL BASED SIMILAR LINKS SUGGESTION

The main objective of our work is to provide Similar links Suggestion for linked open data when the intended link is not available. Due to broken link the paths in the network leading to the practical unavailability of information resulting in simply throwing 404 page not found error message, which will annoy user’s. In that case, similar links are suggested instead of unavailable one. The technique which we have adopted to suggest similar links is a model based approach. The Figure 1 depicts the MBSLS architecture.

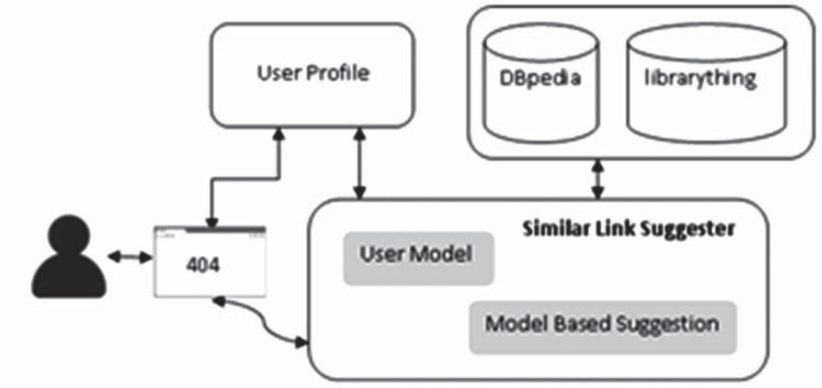

Figure 1. MBSLS Architecture

3.1 User Model

In model based approach, the user profile consists of a model about the user preferences[14][15], i.e., a information about the types of items the user is interested in. The application of Machine Learning techniques is a typical way to achieve the task of learning user profiles[18] in model-based suggestion systems. Creating a model of the user preferences from the user history is a form of classification learning wherein each item has to be classified as interesting or not with respect to the user interest.

3.2 Model Based Suggestion

The core of Model-based suggestion systems mainly lays on text categorization tasks. Machine learning [9] techniques for text categorization has been extensively applied in the field of suggestion systems. Since in our system the similar links to be suggested are resources belonging to semantic datasets, we need to build a model able to deal with such data.

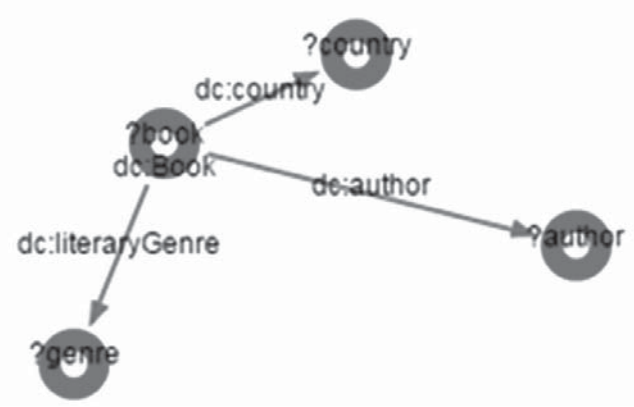

Figure 2(a): Sample Skeleton RDF graph Extracted from DBpedia

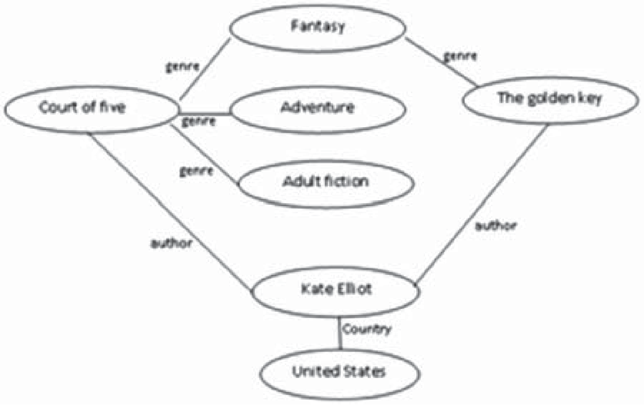

Figure 2(b): Sample Skeleton RDF graph Extracted from DBpedia

According to Figure 2(a), the items to be recommended are the Books and they are described by the nodes they are connected to. The example is about the book domain, others can also use this approach on any domain related to the Web of Data. By exploiting the ontological information encoded via dcterms:books and dcterms:genre properties, we are able to catch implicit relations and hidden information, i.e., information that is not detectable just looking at the nodes directly linked to the item. As an example, if we look at the graph in Figure 2(b), we see that the book court_of_five implicitly have the three genres Fantasy, Adventure and young adult fiction in common. The information discovered by exploiting the taxonomic structure of the genres increases the number of common features between the books.

In our approach the RDF graph is transformed to describe a domain of interest in a feature vector representation[14] [15] that is suitable for the classification task. We have used a bag of words model, dataset are represented by a set of representative keywords (index terms). We adapt the bag of words model in order to deal with RDF triples to obtain a bag of resources model. Taken an item from the collection, for each property we extract all the resources that are linked by the current property to the item and we build an index of resources corresponding to that property (i.e., a property resource-index).

With respect to a given property, each item (i.e., book) is represented by a vector in a multi-dimensional space, where each dimension corresponds to a resource from the vocabulary. For example, referring to Figure 2(b), the resource-index for the property genre is constituted by the resources fantasy, adventure and young adult fiction. Considering all the properties, each item is represented as a unique vector of weights where each weight indicates the degree of association between the item and the resource with respect to a property.

TABLE 1.

TF-IDF Matrix Representation

		Genre		*Author*
Books	Adventure	fiction	fantasy	Kate Elliot
Court of five	0.60	0.60	0.17	0.60
The golden key	0	0	0.17	0.60

We point out that each property resource-index is separated from the others. For all the properties, each item is considered as a unique vector of weights where each weight indicates the degree of association between the item and the resource with respect to a property. These weights are the TF-IDFs [12] and they are computed uniquely for each property resource-index. Table 1 depicts the TF-IDF weights generated from the graph shown in Figure 2(b).

4. SUPPORT VECTOR MACHINE

Our approach requires classifier based on statistical learning with the principle of Structural Risk Minimization, under supervised machine learning technique. Why we chose SVM? Because it excels in text classification tasks and there are more commonalities in user profile while learning for our classification problem. SVMs also provides a two advantages for text classification task: (1) term selection is not frequently required, due to the robust nature of SVMs with respect to over-fitting and dynamically scale up to considerable dimensionalities; (2) no machine and human effort is required during parameter tuning on a validation set is needed. If the decision boundary is not linear, the data is transformed into a higher dimensional space with the help of kernel trick. We have used a RBF kernel because it is the best performing kernel trick with respect to our domain. All our classification is implemented using a WEKA2 SVM (SMO). Our logistic model is built from the outputs of SVM which provides a posterior probability estimates for our classes. Then the outputs are ranked between 0 and 1 from the composed logistic model to suggest the similar links.

5. PERFORMANCE AND EVALUATION

Our evaluation aims to inspect two aspect of the proposed approach (1) analyzing the information contained within the datasets of DBpedia Data which allows the system to achieve the best results in terms of recall and precision[17]; (2) our approach is compared with other systems like content-based, collaborative filtering[1][4] and hybrid[1] ones to analyze the quality in terms of result. To evaluate, initially we aligned the books in the librarything dataset with the books in DBpedia. The SPARQL query to retrieve books information from DBpedia as follows.

```
PREFIX rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
PREFIX rdfs: http://www.w3.org/2000/01/rdf-schema#
PREFIX books: http://dbpedia.org/resource
PREFIX dc: http://dbpedia.org/ontology/
SELECT ?book ?author ?country ?genre
WHERE {
  ?book rdf:type dc:Book;
  dc:author ?author;
  dc:country ?country;
  dc:literaryGenre ?genre
}
```

We have extracted test dataset from the librarything dataset with the constraint of 20 rates per user. Those rates allows us to compute Precision and Recall for values of N in the interval of 1 to 20. Based on binary classifier approach, we considered the user ratings above 3 as like and the others as dislike. The intent of our evaluation is to analyze how well the ontological information present within LOD datasets favour's in suggesting similar links. Our tuned LOD book dataset contains 10,008 books information of different genres and we have made few sample runs on that dataset using our approach to suggest similar links according to user profile.

The Figure 3(a) shows the precision and recall curve based on the property genre after sample run. The red color path clearly indicate the high precision for the property dcterms:genre in terms of Adventure value for suggesting accurate similar links.

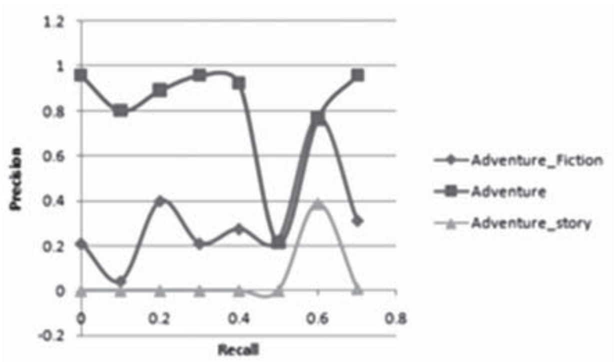

Figure 3(a): Precision and Recall Curves obtained for property

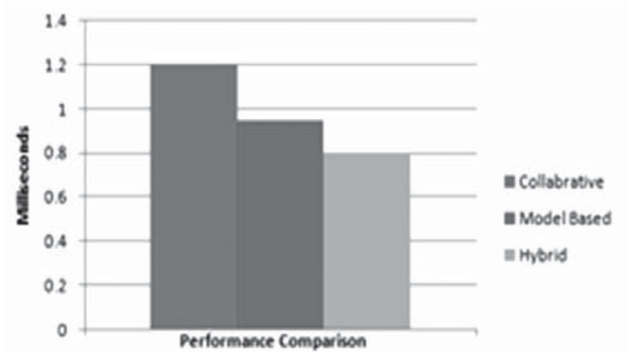

Figure 3(b): Performance comparison between different approaches

Apart from measuring the precision of the result obtained, we have also evaluated the performance of our approach with other approaches. The Figure 3(b) shows the performance evaluation chart which clearly states that our approach clearly excels over collaborative approach in terms of generating similar links within 0.9 milliseconds but lags behind hybrid approach. The reason why our approach performance is degraded over hybrid one is because of cold start[13] problem.

6. CONCLUSION AND FUTURE WORK

In this modern era, the linked open data has grown enormously resulting in vast amount of structured information available globally to the end user's who consumes it. In this paper we have presented, how to handle the broken link by suggesting the similar links using model based approach. The overall accuracy of our system increases because of the ontological nature of the data in LOD datasets. Even though our approach works well in most of the scenario but suffers during cold start situations. We are working hard to improve the performance of our system even under cold start scenario. We are also working on other approach to handle broken link by suggesting similar links from classic web i.e a kind of fusion between semantic web and classic web. Furthermore we are interested in fuzzy based methodologies to handle broken link.

References

1. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors Recommender Systems Handbook. Springer, 2011.
2. P. Lops, M. Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In Recommender Systems Handbook, pages 73-105. 2011.
3. A. Passant. dbrec: music recommendations using dbpedia. In Proc. of 9th Int. Sem. Web Conf., ISWC'10, pages 209-224, 2010.
4. X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. Adv. in Artif. Intell., 2009, 2009.
5. P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos. Moviexplain: a recommender system with explanations. In Proc. of the 3rd ACM Conf. on RSS, pages 317-320, 2009.
6. S. Sen, J. Vig, and J. Riedl. Tagommenders: connecting users to items through tags. In Proc. of 18th WWW, WWW '09, pages 671-680, 2009.
7. Berners-Lee, T. Linked Data - Design Issues. World Wide Web Consortium (W3C) 2006 [cited 2014-04-14].
8. Bizer, C., T. Heath, and T. Berners-Lee. 2009. "Linked Data – The Story So Far." International Journal on Semantic Web and Information Systems no. 5 (3):1-22.
9. Candillier, L., F. Meyer, and M. Boulle. 2007. Comparing State-ofthe-Art Collaborative Filtering Systems. In the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition. Leipzig, Germany: Springer-Verlag.
10. Gantner, Z., S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. 2011. MyMediaLite: a free recommender system library. In the 5th ACM Conference on Recommender Systems. Chicago, Illinois, USA: ACM.
11. Noia, T. D., R. Mirizzi, V. C. Ostuni, and D. Romito. 2012. Exploiting the web of data in model-based recommender systems. In the 6th ACM Conference on Recommender Systems. Dublin, Ireland: ACM.
12. Passant, A. 2010. Measuring Semantic Distance on Linking Data and Using it for Resources Recommendations. In AAAI Spring Symposium Linked Data Meets Artificial Intelligence.
13. Schein, A. I., A. Popescul, L. H. Ungar, and D. M. Pennock. 2002. Methods and metrics for cold-start recommendations. In the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Tampere, Finland: ACM.
14. Z. Eidoon, N. Yazdani, and F. Oroumchian. A vector based method of ontology matching. In Proc. of 3rd Int. Conf. on Semantics, Knowledge and Grid, pages 378-381, 2007.
15. P. Perny and J. Zucker. Preference-based search and machine learning for collaborative filtering: the film-consei recommender system. Information, Interaction, Intelligence, 1:9-48, 2001.
16. C. Wartena, W. Slakhorst, and M. Wibbels. Selecting kwords for content based recommendation. In CIKM, pages 1533-1536, 2010.
17. A. Bellogin, P. Castells, and I. Cantador. Precision-oriented evaluation of recommender systems: An algorithmic comparison. In Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, pages 333-336, New York, NY, USA, 2011. ACM.
18. M. Degemmis, P. Lops, and G. Semeraro. A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation. User Modeling and User-Adapted Interaction, 17(3):217-255, July 2007.
19. B Haslhofer, N Popitsch. DSNotify-detecting and fixing broken links in linked data sets-20th International Workshop on Database and Expert System application 2009. Pages 89-93 - ieeexplore.ieee.org.
20. R. A. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval: The Concepts and Technology behind Search. Addison-Wesley Professional, 2011.