IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 11 (November. 2018), ||V (I) || PP 84-88

Flagged Approach to Detect Broken Links in Linked Open Data

K.Lokeshwaran[1], A.Rajesh[2], P. Nandakumar[3]
1(Research Scholar ,Computer Science and Engineering, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya University, Kanchipuram, India)
2(Principal,Computer Science and Engineering, C. Abdul Hakeem College of Engineering and Technology, Melvisharam,India)
3(Assistant Professor,Computer Science and Engineering, C. Abdul Hakeem College of Engineering and Technology, Melvisharam,India)
Corresponding Author: K.Lokeshwaran

Abstract: The key idea of linked open data (LOD) is to interlink the related contents on the web. Due to the dynamic nature of the web, the contents on the web will change more often. The change in contents on the web leads to broken links. The operations like create, delete, move or update on the web are the major reasons for broken link. Sometimes, if semantic of the contents is altered it will also result in broken link. Semantic broken links are very hard to handle by humans and machines. In this paper we have used well defined flagging approach to identify the semantic broken link. In this approach the semantic modification in the pages are detected using a flagging scheme.

Keywords: Linked open data, Semantic Web, Broken Links, Link Integrity

---

Date of Submission: 11-11-2018 Date of acceptance: 22-11-2018

I. INTRODUCTION

Resource description framework (RDF) is used to publish the structured data in the semantic web. Usually in semantic web the data portrayed in the form of RDF Triples which are exposed to the outside world through different domains like DBpedia, ClinicalTrials etc. Data's are scattered around different data sources9, the related data's have to be interlinked in order to get the proper linked open data4. This interlinking will help the humans and machines to acquire data by navigating through links. Broken link issue is the biggest problem, which restricts the data consumers to retrieve the data's from web of data.

In the web, the entities (resources) are in the form of documents or images. These resources are identified using a uniform resource locator (URI). The operations like create, delete, move or update on the web are the major reasons for broken link4. Popitsch et al., magnifies these operations as11 when new resources are added to the web of data it refers to create operation. If the resources are shifted from one location to another it is called move operation. When resource is no longer accessible it refers to delete operation. Any change made to the already existing resource depicts update operation.

As we all know broken links can be referred in two ways structurally broken link and semantically broken link. If the target resources are not accessible by following the link11 then it is called as structurally broken link. On the other hand if the meaning of the target resource is different from the source then it is called semantically broken link. Till now lot of researcher trying to resolve broken link problem using various methodologies but none has gave productive results. Currently more solutions are available5,11,14 to handle broken link but our aim is to provide more accurate mechanism to identify the semantically broken on web of data. Online self interested group1 has derived few use cases which refers to broken link in which the automated program signals the changes.

The rest of the paper is organized into four sections. In section section2 a review of the related work in this field is discussed. The proposed system and architecture is described in section3. Section 4 is all about experimental setup and evaluation. Future work is discussed in section 5. At the end we have concluded the work proposed in this paper.

II. RELATED WORK

The broken link is one of the biggest challenges in linked open data (LOD). Many researchers interested in broken link problem, however only few have proposed the reliable techniques. Recently the most efficient tool to handle broken link is DSNotify11.

DSNotify11 a framework to handle broken links in linked open data was proposed by Popitsch et-al. He used instance matching strategy to resolve broken link. DSNotify is a well suited tool for both humans and automated agents to identify and resolve broken link. It only deals with structurally broken links but not on semantically broken links. Since web of data is all about huge volume of data, DSNotify uses time interval based blocking mechanism for resource matching and to find out the broken links.

The algorithm proposed by Vesse et-al. depends on extension profile that is utilized to discover information around a URI. The extension profile is a Vocabulary of Interlinked Datasets16 (VoID) that is a group of datasets and linksets specified by the user. In this method, the algorithm discovers proportionate URI utilizing SameAs.org disclosure endpoint and Sindice API. These resources are not accessible for the Web of Data. Moreover, if the noticed dataset does not connect with different datasets these resources won't create any outcomes.

Silk14 a framework to maintain link integrity between the linked open data items and to discover them was developed by Volz et-al. Silk maintains a separate engine to identify links, later those links are evaluated and book keeping is done to maintain the links. Silk is an automated link discovery framework. A special kind of XML based link specification language called Silk-LSL is used to configure the process of automated link generation. SOAP (Simple Object Access Protocol) based WOD-LML (Web of Data-Link Maintenance protocol) is used for the book keeping of links.

Triplify5 a tool to publish data from relational databases. It's an automated tool introduced by Auer et-al. This tool produces the RDF data for the resultant data generated through SQL Queries received from HTTP request. It's also a strategy for distributing current logs of updates that happened inside the particular timespan. Triplify doesn't have automated notification, hence the user agent should pull the notification of update information of their own.

LinQL9 an extension of SQL presented by Hassan Zadeh et-al. LinQL framework consisting of a library of efficient lexical analyzers and similarity functions, and a set of search algorithms for effective and efficient identification of linkage points over Web data. But the LinQL lacks link management technique.

An open source protocol called Pubsubhubbub for loosely interlinked entities in web was developed by Brad Fitzpatrick et-al. It provides instant notification when the links are altered. Notifications are pushed instantly rather than polling.

The method to interlink music dataset was proposed by Reymond12 et-al. His algorithm suggests a similar links based on a good similarity value. Similarities are calculated using the entity and its neighbor. This method doesn't have the link management techniques.

Lui et-al and Li et-al15 used metadata to fix broken links. Every entity has its own particular metadata. Here they proposed, entities were connected through metadata. Consequently, linked data turned into the linked metadata. To resolve broken connections they presented two ideas called "inner ID" and "outer ID". The inner ID is utilized to discover entities inside same data source and outer ID is utilized to get the entity from outside. For every entity there is a mapping between outer and inner IDs. If there is a movement event, the algorithm looks over the metadata to and a proper option for the disengaged entity. On the off chance that the option is discovered, the outer ID is mapped into the new inner ID. However, it is not done in the LOD context, in an attempt to estimate the impact and potential causes for the problem.

III. PROPOSED SYSTEM

Our main objective is to identify the semantically broken links vey precisely. Whenever there is an update in web contents there is more probability of semantic changes. So, the updating of contents should be tracked in order to identify the semantically broken links.

System Overview

The core component of our proposed system is a distributed flag server. We have adopted publish/subscribe system to notify the changes. Our mechanism is somewhat identical to pubsubhubbub. Pubsubhubbub7 uses some additional features in publish/subscribe system where as we have just incorporated very basic model of publish/subscribe system. In our approach the flagging strategy is clearly defined and every time the target entity is updated semantically, it has to be reflagged according to the update.

Distributed Flag Server

The primary role of flag server is to ensure that no flag labels should be similar. So to eliminate ambiguity in flag labels we have used well defined flagging scheme. The flag labels are persisted in a hierarchical structure. This structure ensures the distinctiveness between flag labels. For example instead of using lone flag labels like "lion" it is flagged in a hierarchical way like "\animal\lion". The structured flagging helps the user or agent to understand the semantics of the web to greater extent. The popular site like ACM and Wikipedia uses this structured flagging. Our hierarchical flagging approach is similar to Wikipedia. Wikipedia

uses this hierarchical flagging to identify the related documents3. So the entire hierarchy and the flag labels are persisted in the distributed flag server. Later using these hierarchies the flag labels are categorized into a specific topic. Example hierarchical classification of flag label "banyan" is shown below.

\Root\Tree\Banyan
\Root\organisation\NGO\Banyan

From the above example it is clear that the top levels in the hierarchy depicts the generic classification and the bottom levels denotes the specific classification. The distributed flag server serves the user for two purposes they are flag lookup and flag suggestion

Flag Lookup:

User queries the flag server with flag label in response flag server returns the possible flag hierarchies. From the hierarchies returned by flag server, the user has to search the intended flag hierarchy. The error response from the flag server denotes that no flag hierarchy found. A schematic diagram of flag lookup is shown in Figure1.

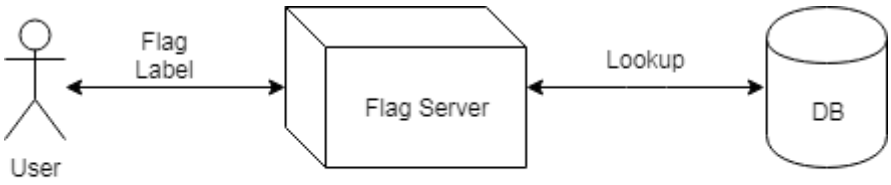

Figure 1: Flag Lookup

Flag Suggestion:

Another role of flag server is to act as flag suggestion engine. Flag server is fed with user content. Based on the user content, the flag server returns the assorted list of flag hierarchies. To suggest the flag hierarchies the flag server runs the document classification algorithm on the user content. We are using a tool called Zemanta for document classification. Zemanta uses the online databases like Wikipedia, Amazon, IMDB etc. to suggest the flags13. Zemanta not only suggest categories it will also suggest well formed keywords.

Notification

The crucial challenge in link management is the notification of update in entities. So, we have used publish/subscribe model to address this challenge. This model comprises of three major participants broker, publisher and subscriber. There is no limit for number of participants. Subscribers are source dataset and target dataset as a publisher. The broker receives a subscription message from source dataset whenever link is created. The format of subscription message will be <source,target> pair. All the brokers shares the subscription information to avoid single point of failure.

If the target document contents updated, to ensure the update target self checks the changes by consulting with flag server. On the process if target confirms the changes, a publish message is sent to the related broker. The current flag set is the content of the publish message. Later this message is forwarded by broker to the sources of the related target as a notification message. Using this message the source self checks the document for changes and can probe to correct the links.

According to hierarchical flagging scheme, tag should be iterated from root to topic along the path. To identify the similarity between the flags the publisher can use any mechanism. One well known mechanism is the ratio of number of elements in the hierarchy matches with another tag hierarchy. We have incorporated cosine similarity measure along with the similarity threshold value. So, if the similarity measure is below the threshold then it clearly indicates the semantic changes and publish message is sent to a broker. Scalability factor of the system is determined by the communication model and the amount data it can handle. Our system scalability is ensured by publish/subscribe model and distributed data sources.

IV. EXPERIMENTAL SETUP AND EVALUATION

As stated in Section3, the proposed framework comprises of a flag server N number of publishers, subscribers brokers and subscription details are stored in distributed database. The consolidated architecture of our system is shown in Figure 2. Separate services for notification and subscription are implemented in broker network. As like broker network for publisher and subscriber also separate services are implemented for communication. To evaluate we have implemented a system based on the architecture. Also to represent entities

we have implemented individual RESTful web services. Instead of SOAP10 we have opted for RESTful web services because of its efficiency and its simple communication model. Jersey6 a java based RESTful web service framework is used for implementation. For data persistence MYSQL database is used. Zemanta13 linked data based recommender system is used as a tag server.

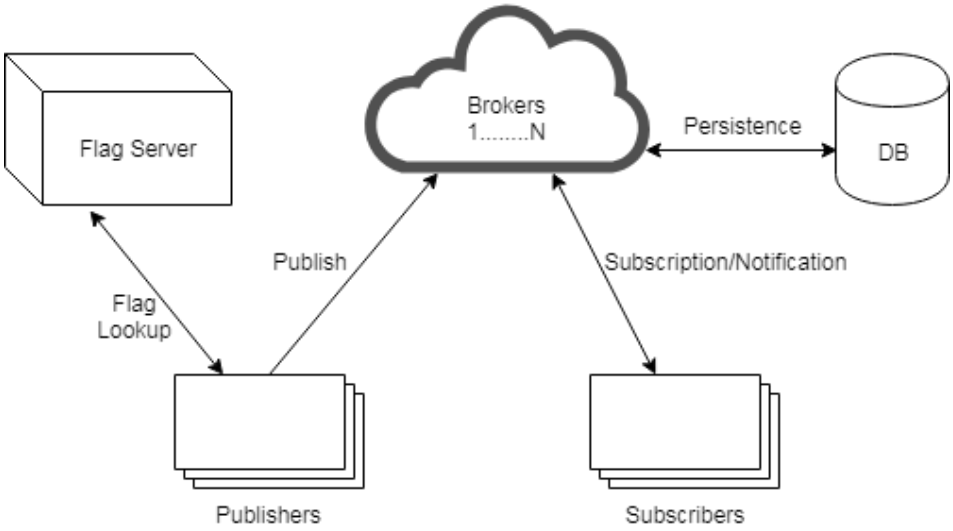

Figure 2: System Architecture

We have measured the accuracy of our system in terms of broken links detection. So, we have created a set of subscription from some randomly selected source and target URI's. In order to verify our system we have collected a dataset sample8 UCI machine learning repository. We have simulated the system by randomly shuffling the content of the documents. The dataset is segregated into two types, one for the specific area and another for the diverse fields. Each dataset contains 24 abstracts. We ran our simulation test on both the datasets. The result is shown in the Figure 3. It clearly shows that our broken link detection system is good for both specific and diverse fields.

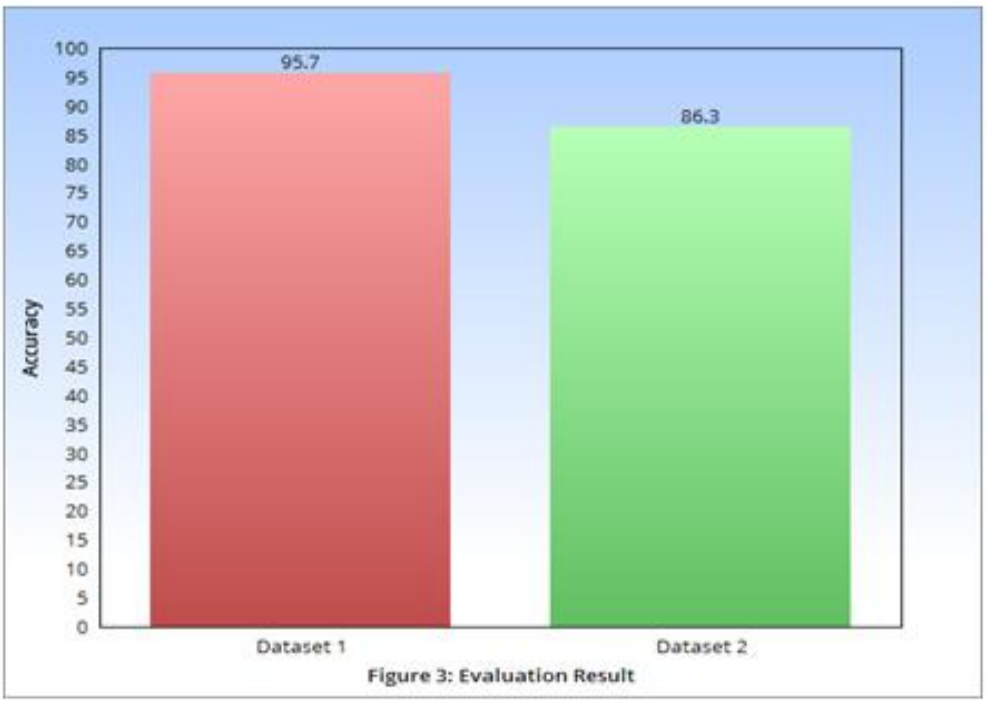

Figure 3: Evaluation Result

V. FUTURE WORK

We are still evaluating our system on larger dataset. Automated machine agents struggles in detecting broken links, so human assistance is required to support machine agents. We are trying to eliminate the human assistance to make the system fully automated. Our system only detects broken links. In the future, we intend to assess the proposed system with different datasets from various domains and will attempt to fix the semantic broken connections.

VI. CONCLUSION

We have shown a method to detect broken links in LOD. Our system is not a fully automated one but still it proves its worth in terms of high accuracy. This is possible only because of doing the semantic check during content update. Flagging approach used in this system is to improve the overall accuracy. The system which we designed is scalable in nature. The biggest advantage of our system is event notification, which is achieved through loosely coupled publish/subscribe model. The scalability of a communication system is ensured through RESTful web services. This system is a good plug-in for current link integrity solutions.

REFERENCES

[1]. DatasetDynamics - W3C Wiki. http://www.w3.org/wiki/DatasetDynamics.
[2]. Document classification - wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Document_classification.
[3]. Wikipedia:categories, lists, and navigation templates. http://en.wikipedia.org/wiki/Wikipedia:Categories,_lists,_and_navigation_templates.
[4]. H. Ashman. Electronic document addressing: dealing with change. ACM Comput. Surv., 32(3):201–212, Sept. 2000.
[5]. S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, and D. Aumueller. Triplify: light-weight linked data publication from relational databases. In Proceedings of the 18th international conference on World wide web, WWW '09, pages 621–630, New York, NY, USA, 2009. ACM.
[6]. T. Berners-Lee. Linked data - design issues. http://www.w3.org/DesignIssues/LinkedData.html, 2006.
[7]. B. Fitzpatrick and B. Slatkin. Pubsubhubbub core 0.3 – working draft . http://pubsubhubbub.googlecode.com/svn/trunk/pubsubhubbub-core-0.3.html, 2010.
[8]. Frank and A. Asuncion. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2010.
[9]. O. Hassanzadeh, A. Kementsietsidis, L. Lim, R. J. Miller, and M. Wang. A framework for semantic link discovery over relational data. In Proceedings of the 18th ACM conference on Information and knowledge management, CIKM '09, pages 1027–1036, New York, NY, USA, 2009. ACM.
[10]. G. Mulligan and D. Gracanin. A comparison of soap and rest implementations of a service based interaction independence middleware framework. In Winter Simulation Conference, WSC '09, pages 1423–1432. Winter Simulation Conference, 2009.
[11]. N. P. Popitsch and B. Haslhofer. Dsnotify: handling broken links in the web of data. In Proceedings of the 19th international conference on World wide web, WWW '10, pages 761–770, New York, NY, USA, 2010. ACM.
[12]. C. Raymond, Yves Sutton and M. Sandler. Automatic interlinking of music datasets on the semantic web. In Proceedings of the 1st workshop about linked data on the web, LDOW '08. Proceedings of the 1st workshop about linked data on the web, 2008.
[13]. Tori and T. Solc. Zemanta service. http://developer.zemanta.com/media/files/docs/zemanta_api_companion.pdf.
[14]. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data.In Proceedings of the 8th International Semantic Web Conference, ISWC '09, pages 650–665, Berlin, Heidelberg, 2009. Springer-Verlag.
[15]. Liu, F. and Li, X., Using Metadata to Maintain Link Integrity for Linked Data Proceedings of the 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing (2011), Vol. 14, pp.432-437.
[16]. Vesse, R., Hall, W. and Carr, L., Preserving Linked Data on the Semantic Web by the application of Link Integrity techniques from Hypermedia, LDOW (2010).

K.Lokeshwaran. "Flagged Approach to Detect Broken Links in Linked Open Data." IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 11, 2018, pp. 84-88.