Data Integration and Analysis in music domain: leveraging Semantic Web and Linked Open Data

Viral Patel, Yogesh Patel

Rollwala Computer Center, Gujarat University, Ahmedabad, India

Abstract: *This paper describes how the linked data growth on Internet and development of linked data application is awkward. This paper describes how SPARQL query work on open linked data cloud using semantic web. We have built the system that will collect the standard information about the music. The information is collected from different data source available in linked open data cloud(Freebase, DBPedia, Music Brainz, WikiData and BBC music) and linked all the data sets through linked data. Our results will give suggestion about the data source which has more particular information. It will work like the routing table. Our system results will tell about what information user can get better and more from which data source. It will give all the links in it about where to find information about music work.*

Keywords: *linked data, sparql, data sources, properties, mapping.*

I. INTRODUCTION

Linked data is best practice for publishing data on the web. Linked data simply create the links between the different data sources available on the Web. Data from different domain, self-describing are linked together. Linked data published the data on the Web such a way that it can be machine readable as well as human readable. Linked data is linked to other external data sets and can be linked from external data sets. The main purpose of Linked data is to establish the connection through chains of URIs that refers to same individual data sets.
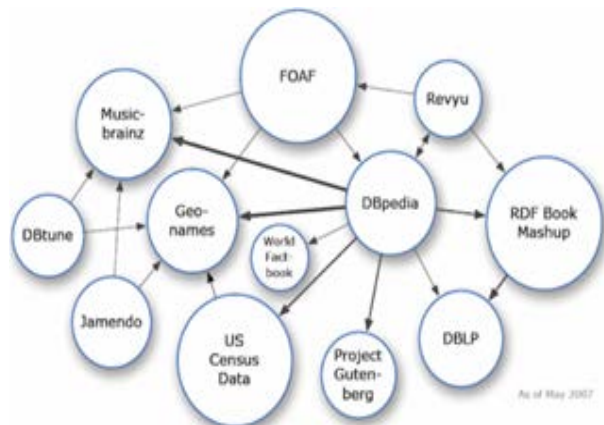

Fig 1.1 Linked data cloud (2007)

Fig 1.1 shows the Linked data cloud 2007. In the linked data cloud diagram nodes which are published data sets are connected with other data sets through URIs with the links which are RDF triples. For example here the link from DBPedia and MusicBrainz in which DBPedia is not only representing information from DBpedia's data set but it also representing similar information from MusicBrainz data set.

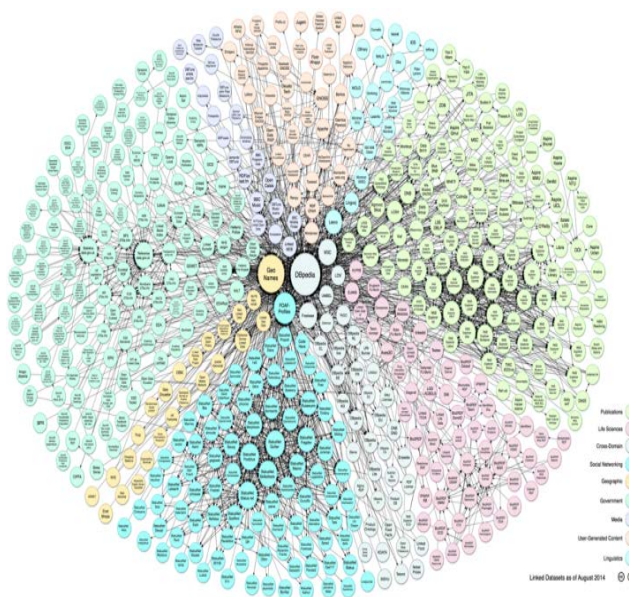### A. Growth of the Linked Data

Fig 1.2 Linked open data cloud (*2014-08-30*)

Fig. 1.2 shows the diagram of the Linked data cloud for 2014 and one can compare Fig 2 with the Fig 1 and one can see the exponential growth that linked data made during these years.

B. Linked Data Principle

According to Tim Berners-Lee he describes four simple rules in 2006 to publish data on the Web. The main goal of these rules was that as many people will follow these principles it is more convenience for other people to use these data.

Here are the principles

1. Instead of Names use URIs to identify things.

2. Use HTTP URIs so that people can look up those names.

3. Provide useful information with the help of standard like (RDF,SPARQL) when people tries to search the URIs.

4. To discover more thing include links to other URIs.

By using the URI to identify the all information that we needed and also gave us the source where to look up for the more information. Only thing that we need to take care is that the URIs is not supposed to be broken links otherwise we will not get benefits of using URIs. We can gain more benefit if link given in the URIs can lead to other resources and from there we can fine more information that are related to the URIs.

C.   Querying Linked Data

SPARQL is fundamentally query language from which information can be retrieve from datasets. SPARQL is a semantic web technology. RDF triple, RDFS, Graph Pattern, SPARQL endpoints etc. are various SPARQL terminology that can be used to information retrieval. There are mainly four types of SPARQL query which are ASK, SELECT, CONSTRUCT, and DESCRIBE.

II.   EXPERIMENT CONDUCTED

For Experiment we have used Programming Language as JAVA. We have used MYSQL as local database and used Apache JENA library for sparql querying on sparql end-points and Retrieving Results in any format. Here we take three open datasets dbpedia, linkbrainz (MusicBrainz) and bbc.uk.

A.   DBPedia:

DBPedia was the project created by the people of the Free University of Berlin and the University of the Leipzig and made publicly in 2007. DBPedia is database that associated with the Wikipedia resources. DBPedia was created to extract structured information from the Wikipedia resources and then available on the World Wide Web. DBPedia allow others to use database for retrieving and adding information from it. DBPedia has many properties that helps user to semantically query relationship between Wikipedia resources and linked to other datasets.

Fig. 2. DBPedia Properties

These are some music related DBPedia properties that helps user to retrieve data from DBPedia sparql endpoint and also helps if any user wants to include some information into datasets. User can use any of these properties to retrieve information about the musical artist. DBPedia has mostly biographic information about the musical artist. For example user wants to know the birthdate of any musical artist then user needs to use "dbpedia-owl:birthDate" or "dbpprop:birthDate". So this way user can use any properties to get desire information from DBPedia using sparql query.

DBPedia has sparql end-point "http://dbpedia.org/sparql" where anyone can perform hands on. First using DBPedia sparql end-point. We have built java program that retrieved around 88000 musical artists using sparql query and stored it in our local database (i.e. MySQL). We have stored all URI's of the musical artist in our Database. Now we have table that has all URI of the musical artists. From this table we have retrieved all the information about the musical artist like name, role, birth date, birthplace, genre, lccnid, viafid, isniid etc. and stored it in our database using java program. Now again using musical artist table we have created program that retrieved all the songs of that musical artist one by one and stored in in our database. Then we have created another java program that retrieved all the information about the songs like name of artists who made the song, band who performed, album in which the song belongs, release date of the song, under the label in which the song is released, runtime of the song, format of the song, genre of the song, producer and writer of the song etc. and stored it in our database.

B.   MusicBrainz:

MusicBrainz was created to maintain open content music datasets. First MusicBrainz was created to make compact disk databases but then it became structure online music

datasets. MusicBrainz is open content datasets so any user can import and export musical related information using written guidelines of the community. MusicBrainz have more music related information like artist, their work done, relationship with other artists, albums recorded, track title, length etc. MusicBrainz have many properties that helps user to retrieve information about the musical artist and their work done.

MusicBrainz		
Artist	**Album**	**Track**
id	db:vocab/album_attributes	dc:date
db:vocab/artist_quality	db:vocab/album_quality	dc:title
db:vocab/artist_tag_count	db:vocab/album_script	mo:length
db:vocab/artist_type	db:vocab/albummeta_discids	mo:musicbrainz
db:vocab/sortname	db:vocab/albummeta_puids	mo:track_number
mo:musicbrainz	db:vocab/toc	rdf:type
tags:taggedWithTag	db:vocab/tracks	rdfs:label
rdf:type	dc:date	foaf:maker
rdfs:label	dc:language	mo:published_as
owl:sameAs	dc:title	mo:track
foaf:name	mo:musicbrainz	
event:agent	mo:release	
mo:background_singer	mo:track	
mo:lead_singer	rdf:type	
mo:performer	rdfs:label	
mo:singer	foaf:maker	
rel:collaboratesWith	foaf:page	
foaf:maker	mo:published_as	

Fig. 2.2 MusicBrainz Properties

These MusicBrainz properties helps user to retrieve information from MusicBrainz using MusicBrainz sparql endpoint. User can also add any information using these properties. For example if any user wants to know the track length and what type of that track is then using properties "mo:length" and "rdf:type"user can find out. MusicBrainz has sparql endpoint "http://dbtune.org/musicbrainz/snorql/" from there user can do some hands on to know how endpoint works. The process of retrieving information that we have done on DBPedia now we are doing on MusicBrainz. Like DBPedia we are retrieving musical artist from MusicBrainz stored it in our database. Then all the songs of the musical artists and all the details of the songs we have stored it in our database.

III. RESULT ANALYSIS

This table specifies the properties of Musical artists. It shows that which property is available in which data sets. For example name, track, album and id these properties are available in all 3 data sets (DBPedia, MusicBrainz and BBC). While some properties like date of birth, place of birth, lccn, isni are available in only DBPedia. We have only taken properties which are working on sparql end point.(i.e. http://lod.openlinksw.com/sparql/). We discarded other properties which did not worked on sparql end point .BBC

music has many properties but here we took only two properties that are working on http://lod.openlinksw.com/sparql/. This is sparql end point for BBC music; we took only two because bbc.uk has not sparql endpoint in working condition.

Table 1. Music Professional Properties

Music Professional	DBPedia	MusicBrainz	bbc.co.uk
Name	YES	YES	YES
Date of Birth	YES	NA	NA
Place of Birth	YES	NA	NA
Genre	YES	NA	NA
Credit	YES	NA	YES
LCCN	YES	NA	NA
Track	YES	YES	YES
Album	YES	YES	YES
Event	NA	YES	YES
ISNI	YES	NA	NA
VAIF	YES	NA	NA
ID	YES	YES	YES

Table 2. Musical Tracks Properties

Musical Track	DBPedia	MusicBrainz	bbc.co.uk
Name	YES	YES	YES
Date of Publish	YES	NA	NA
Place of Publish	YES	YES	NA
Genre	YES	NA	NA
Album	YES	YES	NA
Composer	YES	NA	NA
Lead Artist	YES	YES	YES
Producer	YES	NA	NA
Song Writer	YES	NA	NA
Record Company	YES	NA	NA
Run Length	YES	YES	NA
ID	YES	YES	YES

This table specifies the properties of Musical tracks. It shows that which property is available in which data sets. For example name, place of publish, album, lead artist and run length these properties are available in all 3 data sets (DBPedia, MusicBrainz and BBC). While some properties are only available in DBPedia.

A.　Statistics about current Datasets (music professional)

Table 3. Music Professional Statistics

	DBPedia	MusicBrainz	bbc.co.uk
Total #Artist	88841	~90K	~90k
Total #Properties	147	45	2
Total #null who	49067 (55.23%)	NA	NA
Total #Null genre	26978 (30.36%)	NA	NA
Total #null date of birth	52066 (58.60%)	NA	NA
Total #Null place of birth	64830 (72.97%)	NA	NA
Total #Null lccnid	87974 (99.02%)	NA	NA
Total #Null viafid	72486 (81.59%)	NA	NA
Total #Null isniid	88791 (99.94%)	NA	NA

This table specifies that approximately we have collected 88841 DBPedia musical artists, ~90,000 MusicBrainz musical artists and ~90,000 BBC musical artists.

Total number of properties describe that DBPedia has 147 properties, MusicBrainz has 45 properties, and BBC has 2 properties (only we have included) about music professional.

Total number of who describes the role of musical artists where artist is a lead singer, guitarist, drummer etc. It shows that 55.23% values are null in DBPedia where MusicBrainz and BBC do not have this values.

Total number of date of birth describes the birth date of musical artists. DBPedia has 58.60% null values in it where MusicBrainz and BBC do not have this value.

Total number of place of birth describes the birth place of musical artists. DBPedia has 72.97% null values in it where MusicBrainz and BBC do not have this value.

Total number of lccnid describes the lccn id of musical artists. DBPedia has 99.02% null values in it where MusicBrainz and BBC do not have this value.

Total number of viafid describes the viaf id of musical artists. DBPedia has 81.59% null values in it where MusicBrainz and BBC do not have this value.

Total number of isniid describes the isni id of musical artists. DBPedia has 99.94% null values in it where MusicBrainz and BBC do not have this value.

B.　Statistics about current Datasets (songs)

Table 4. Musical Track Statistics

	DBPedia	MusicBrainz	bbc.co.uk
Total #Artist	88841	~90K	~90k
Total #Properties	147	45	2
Total #Songs with NULL	117,864	146538 (~<5000 Artists)	126,346
Total #Null Songs	80675(90.80% Artists have no songs Available)	938	75786(84% Artists have no songs Available)
Total # Songs	37189	145600	50560
Total #Null Names in Songs	1198 (3.22%)	195 (0.13%)	200 (0.37%)
Total #Null Artist in Songs	.36%)	195 (0.13%)	200(0.37)
Total #Null Band in Songs	453 (1.21%)	NA	NA
Total #Null Album in Songs	6951 (18.69%)	15%)	NA
Total #Null Record label in Songs	4959 (13.33%)	NA	NA
Total #Null Record Date in Songs	(7.30%)	NA	NA
Total #Null Runtime in Songs	(14.42%)	(1.15%)	NA
Total #Null Format in Songs	10645 (28.62%)	NA	NA
Total #Null Genre in Songs	(6.69%)	NA	NA
Total #Null Producers in Songs	13005 (34.97%)	NA	NA
Total #Null Writer in Songs	11753 (31.60%)	NA	NA
Total #Null Certification in Songs	34535 (92.86%)	NA	NA
Total #Null Track number in Songs	NA	1683 (1.15%)	NA

This table displays the properties of songs.

Total numbers of songs available in DBPedia are 117864, MusicBrainz are 146538 and BBC music's are 126346.

Total number of null songs that means numbers of artist that does not have song values are DBPedia 80675 (90.80%), MusicBrainz 938 and BBC 75786 (84%).

Total numbers of songs with details available are DBPedia 37189, MusicBrainz 145600 and BBC 50560.

Total numbers of null names in DBPedia are 1198 (3.22%), MusicBrainz 195 (0.13%) is and BBC music's are 200 (0.37%).

Total numbers of null artists in DBPedia are 452 (1.36%), MusicBrainz 195 (0.13%) is and BBC music's are 200 (0.37%).

Total numbers of null bands in DBPedia are 453 (1.21%), MusicBrainz and BBC music do not have this property.

Total numbers of null album in DBPedia are 6951 (18.69%), MusicBrainz 1683 (1.15%) are and BBC music do not have this property.

Total numbers of null record labels of songs in DBPedia are 4959 (13.33%), MusicBrainz and BBC music do not have this property.

Total numbers of null record date of songs in DBPedia are 2715 (7.30%), MusicBrainz and BBC music do not have this property.

Total numbers of null runtime of songs in DBPedia are 5363 (14.42%), MusicBrainz 1683 (1.15%) are and BBC music do not have this property.

Total numbers of null format of songs in DBPedia are 10645 (28.62%), MusicBrainz and BBC music do not have this property.

Total numbers of null genre of songs in DBPedia are 2488 (6.69%), MusicBrainz and BBC music do not have this property.

Total numbers of null producers of songs in DBPedia are 13005 (34.97%), MusicBrainz and BBC music do not have this property.

Total numbers of null writers of songs in DBPedia are 11753 (31.60%), MusicBrainz and BBC music do not have this property.

Total numbers of null certification of songs in DBPedia are 34535 (92.86%), MusicBrainz and BBC music do not have this property.

Total numbers of null track number in MusicBrainz are 1683 (1.15%), DBPedia and BBC music do not have this property.

C.   Statistics about Properties

Table 5. Statistics about Properties

Total #properties in dbpedia	147
Total #properties in musicbrainz	45
Total #properties in bbc	2
Total #properties in wikidata	436
Total #properties overlap between dbpedia and musicbrainz	12
Total # properties overlap between dbpedia and bbc	2
Total # properties overlap between musicbrainz and bbc	2
Total # properties recommended to musicbrainz from dbpedia	11
Total # properties recommended to bbc from dbpedia	21
Total # properties recommended to dbpedia from wikidata	22]

As we can see from the table that properties of different data sets like DBPedia have 147, MusicBrainz have 45, BBC have 2 (only we took), wikidata have 436. From above table we can determine that there are many properties that datasets can recommend to each other so all data sets can have more information. For example DBPedia can recommend 11 properties to MusicBrainz, DBPedia can recommend 21 properties to BBC Music, and Wikidata can recommend 22 properties to DBPedia. There are some properties that are overlap with each other. For example properties overlap between DBPedia and MusicBrainz are 12, between DBPedia and BBC are 2 and between MusicBrainz and BBC are 2.

D.   Property Recommendation

Here from the given image we can determine that some properties like dbpedia-owl:birthPlace, dbpedia-owl:genre etc. from DBPedia are recommend to MusicBrainz and BBC. Properties like dbprop:credits, dbpedia-owl:runtime etc. are recommend to wikidata. Properties like LCNAF, IMDB and LCCN etc. from wikidata are recommending to

DBPedia, MusicBrainz, and BBC. All properties recommendations are given in fig 3.1.

Fig. 3.1 Property Recommendation

IV.   MAPPING

Fig. 4.1 Mapping of Data Sets

This Fig 4.1 shows the master schema mapping that will connect all the property of relevant data sets. In this mapping we have mapped musical artist's property with all data sets. We can determine that which Data sets have what kind of property and which data sets has more number of properties. From this mapping we can determine which data sets user will choose for getting required data sets.

A.   Result of Mapping Data Sets

This is the result of our mapping data sets where we have linked all five data sets (DBPedia, MusicBrainz, Freebase, BBC, and Wikidata) with each other. We used same as property that linked all data sets. First we linked BBC music and MusicBrainz with each other. Then we have linked BBC music with DBPedia and then we have linked DBPedia with Wikidata and Freebase.

We have successfully integrated BBC music and MusicBrainz 99% with each other. But when we integrated BBC music with DBPedia we could only connect around

23,000 artists. This thing clearly describe that lots of artist yet need to be linked into linked open data cloud.

Fig. 4.2 Mapping Data set

V.   CONCLUSION

From the various data sources we tried to get information of musical artists and that work. While retrieving the information we can conclude that some data source like DBPedia has more number of properties available but these properties need to be filled with values.

If a user wants to find the biographic information about the musical artist then user can visit DBPedia or WikiData. While another data source like MusicBrainz has less biographic properties but it has more properties about the songs and albums.

WikiData and Freebase has properties available but they did not provide any endpoint so we cannot derive any result from it.

In our analysis we included all mapping links of data source so we can give an idea that on which link user should go to retrieve user's desirable information. Instead of searching directly on Internet we are asking user to search through our database so we can suggest user that particular properties or information can easily available on particular data sets.

VI.   FUTURE WORK

One can make interface that helps user to search information about music related information. Our work (analysis) will give suggestion about the data source which has more particular information. It will work like the routing table. It will show that which information user can get better and more from which data source. It will give all the links in it about where to find information so someone can use these results to make search engine portal.

R<small>EFERENCES</small>

[1] Pattuelli, M. Cristina. "Mapping people-centered properties for Linked Open Data."*Knowledge Organization* (2011).

[2] Bizer, Christian, Tom Heath, and Tim Berners-Lee. "Linked data-the story so far."*International journal on semantic web and information systems* 5.3 (2009): 1-22.

[3] www.euclid.com.

[4] http://lod-cloud.net.

[5] S. Singh and N. Singh, "Big Data Analytics", 2012 International Conference on Communication, Information & Computing Technology Mumbai India, IEEE, October 2011.

[6] Davis, Ian, "An Introduction to RDF", http://research.talis.com/2005/rdf-intro/.

[7] Siddharth Gupta , Narina Thakur, "Semantic Query Optimization with Ontology Simulation" , International journal of web & semantic Technology (IJWesT) Vol.1 , Num 4 , October 2010.

[8] G. Klyne and J. J. Carroll, editors. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, 10 February 2004. Available from http://www.w3.org/TR/rdf-concepts/.

[9] E. Prud'hommeaux and A. Seaborne (2008) "SPARQL Query Language for RDF". Published on-line at http://www.w3.org/TR/rdf-sparql-query/.

[10] Bizer, c., cyganiak, R., Heath, T. (2007). How to publish Linked Data on the Web. Retrieved June 14, 2009, http://www4.wiwiss.fuberlin.de/bizer/pub/LinkedDataTutorial/.

[11] cyganiak, R., Bizer, c. (2008). Pubby - A Linked Data Frontend for SPARQL Endpoints.

[12] Bizer, C., Schultz, A.: The R2R Framework: Publishing and Discovering Mappings on the Web. 1st International Workshop on Consuming Linked Data (COLD 2010), Shanghai, November 2010.

[13] Haslhofer, B., Schandl, B. (2008). The oAi2LoD Server: Exposing oAi-PMH Metadata as Linked Data. Proceedings of the 1st Workshop about Linked Data on the Web (LDoW2008).