GRAPH-DRIVEN ANALYSIS AND VISUALISATION OF FREEBASE SCHEMA AS A DIRECTED WEIGHTED GRAPH

Mahmoud Elbattah [1], Mohamed Roshdy [2], Mostafa Aref [3], Abdel-Badeh Salem [4]
College of Engineering & Informatics, National University of Ireland [1], Faculty of Computer and Information Sciences Ain shams University, Cairo, Egypt [2,3,4]

ABSTRACT: *Freebase can be considered as one of the largest sources in the Linked Open Data (LOD) cloud. The paper adopts a graph-driven approach to analyse and visualise Freebase schema. Firstly, the paper demonstrates the extraction process of Freebase schema through stepwise procedures that maintain the schema structure. Secondly, the extracted types and relationships are utilised to represent Freebase schema as a directed weighted graph. Furthermore, the schema graph is used to conduct graph-based analysis and visualisation that can describe and summarise the schema structure and organisation of Freebase data.*

KEYWORDS: Linked Open Data, Freebase, Data Visualisation.

INTRODUCTION

Freebase can be appropriately described as a collaboratively built, graph-shaped database of structured general human knowledge [1]. Freebase includes a great diversity of structured data imported from different sources such as Wikipedia, MusicBrainz and WordNet [2]. Unlike other LOD-databases, DBpedia for example, users of Freebase can directly edit and change the data and the structuring schema [3]. Therefore, Freebase enables public read/write access allowed through an HTTP-based query APIs using the Metaweb Query Language (MQL).

Freebase data structure is based on networked graphs where nodes, representing entities, are connected by edges [7]. Figure (1) illustrates how Freebase schema model is organized where the schema is organized around the following objects:

- Domain: A collection of types which share a namespace.
- Type: Denotes an Is-A relationship about a topic.
- Property: Define a Has-A relationship between the topic and the value of the property.
- Category: A grouping of related domains.

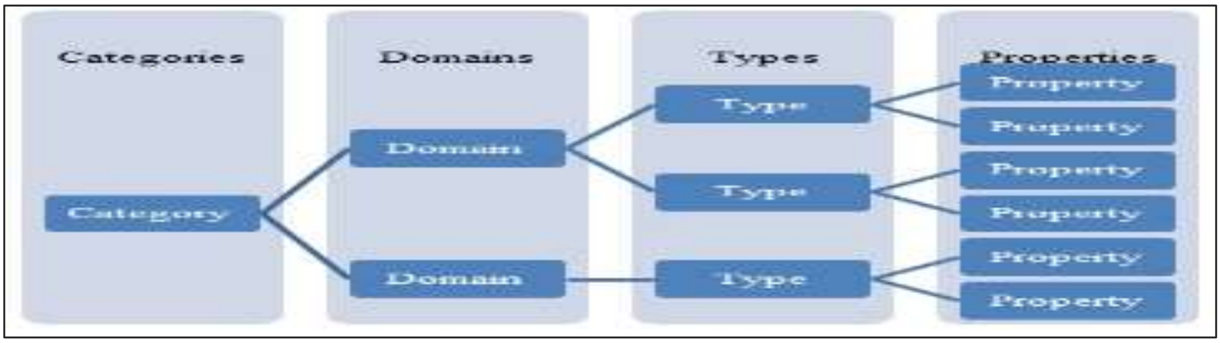

Figure 1. Hierarchy of Freebase schema.

Freebase currently contains more than 125,000,000 tuples, more than 4000 types, and more than 7000 properties [4]. The massive amount of Freebase data raises an inevitable demand for effective data visualisation techniques.

The paper adopts a graph-based approach for the analysis and visualisation of Freebase schema. Firstly, the extraction process of Freebase schema is explicitly presented, since Freebase schema is not easily accessible via downloadable data dumps, according to the best of the authors' knowledge. Secondly, the extracted schema objects are utilised for constructing the schema graph. Eventually, the schema graph is employed for conducting graph-based analysis and visualisation that can describe or summarise how Freebase data are organised around the schema model.

Specifically, we claim the following contributions:

- Explicitly demonstrating the required step-wise procedures for extracting Freebase schema, such that maintain the schema model.
- Proposing a methodology that utilises the "included-type" relationships and "instance counts" of Freebase schema to construct a directed weighted schema graph.

LITERATURE REVIEW

"Thinkbase" [5][6] was a significant endeavours for developing a visualisation and exploration tool of Freebase data. Thinkbase was designed and implemented to extract the contents and semantic relationships from Freebase, and interactively visualise the semantic graphs extracted. For an entity with rich properties, the graph could have hundreds of nodes and become too cluttered on screen. On the other hand, "GraphCharter" [14] combined graph browsing with query to overcome the limitation of visual inspection. GraphCharter was presented as a general graph exploration method for large-scale semantic graphs that enabled graph browsing with the ability to query. GraphCharter used Freebase as case study of high density semantic graphs. However, the visualisation of Freebase schema itself has been slightly addressed in literature, according to the very best of the authors' knowledge. We believe that a good understanding of the large-scale schema of Freebase can enhance the querying experience of the massive amount of Freebase data. Therefore, the major attention in this paper is drawn to the Freebase schema in particular. Graph-driven approach is adopted for analysing and visualising Freebase schema.

METHODOLOGY

This section demonstrates in detail the adopted approach for the extraction, analysis and visualisation of Freebase schema, as follows:

Extraction of Freebase Schema

Although Freebase provides downloadable RDF-formatted data dumps, Freebase schema itself is not easily available like the data dumps. Therefore, Freebase schema had to be extracted as explained in this section. The main components of the schema extraction process, illustrated in figure (2), are as follows:

1. MQL Query: The JSON-based query syntax provided by Freebase.

2. HTTP Request API: The MQL queries were delivered to Freebase through API HTTP requests. Freebase supports two types of APIs for reading purposes, which are “MQLRead” and “Search”. However, “MQLRead” was only used since it supports complex full queries while “Search” is more parameter-based.
3. HTTP Response: Freebase returned the query result through HTTP response as well. The results were JSON-encoded, referred as “envelope” according to Freebase [9].
4. JSON Parser: A JSON parser should be used to deserialise the JSON-formatted objects of query results. JSON.Net library [10] was used for that purpose.
5. Extracted Schema: The extracted schema of Freebase was stored in a relational database.
6.

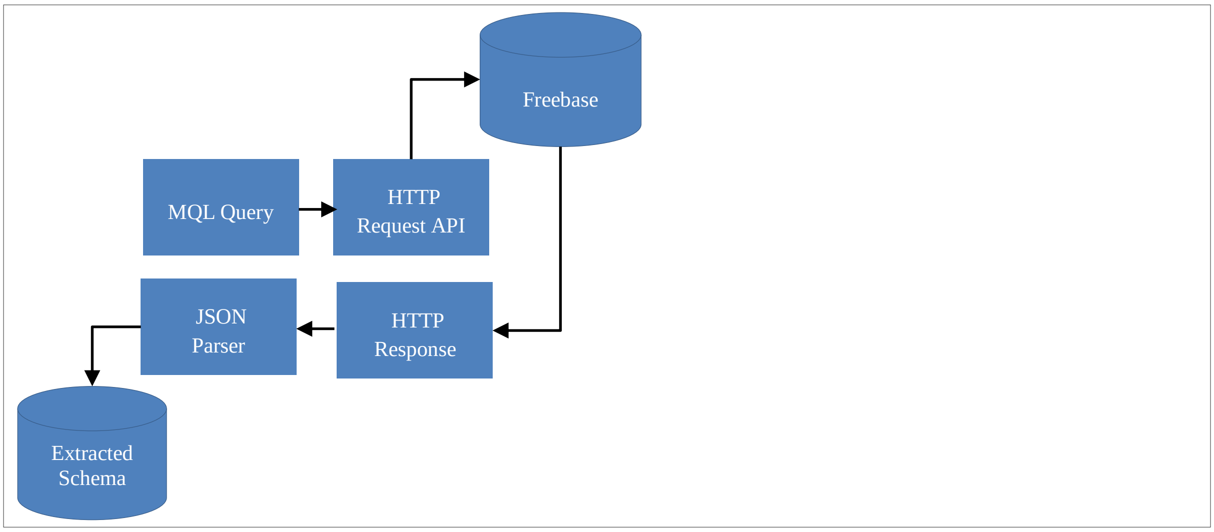

Figure 2. Main components of the schema extraction process.

Extraction Procedures of Freebase Schema

The Freebase schema was extracted by stepwise procedures that maintained the organization and structure of the schema model. Figure (3) lists the order of the five steps conducted.

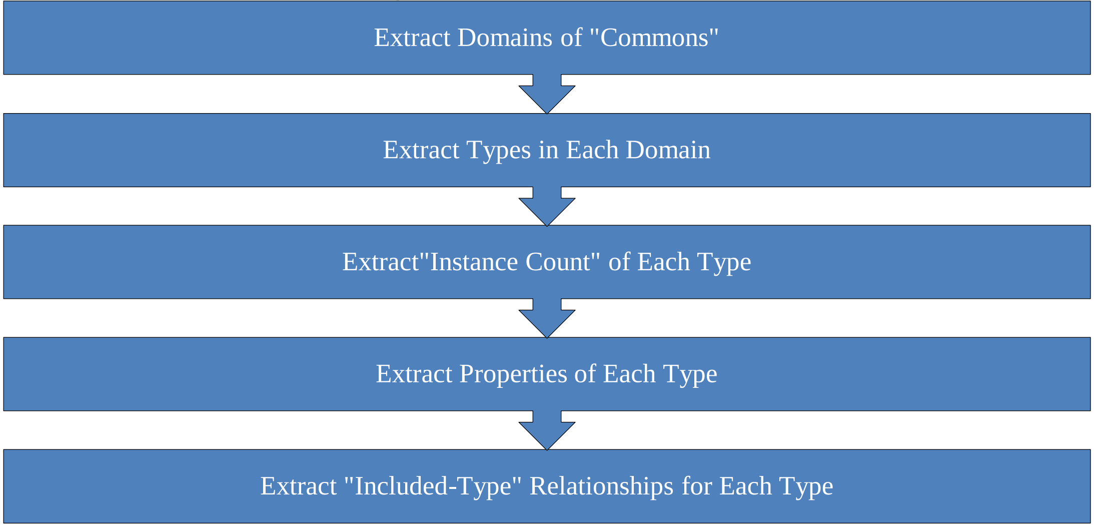
Extract Domains of "Commons"
Extract Types in Each Domain
Extract"Instance Count" of Each Type
Extract Properties of Each Type
Extract "Included-Type" Relationships for Each Type

Figure 3. Stepwise procedures of the schema extraction process.

The scope of the schema extraction was limited by the “Commons” category. The Commons category was selected since its types and properties should meet a certain high standard, according to Freebase [11]. Table (1) summarizes the extracted schema objects. In addition, Table (2) presents the MQL queries necessary for each procedure of the extraction procedure.

Table 1. Summary of the Extracted Objects of Freebase Schema

Freebase Schema Element (Commons Category)	Number of Extracted Objects
Domains	82
Types	2,210
Properties	6,851
Included Types Relationships	2,845

Table 2. List of MQL Queries Associated with Each Extraction Procedure

Procedure	MQL Query Text	Example(s) of Query Result	Remarks
Extract Domains of Commons	[{ "id": null, "name": null, "type": "/freebase/domain_profile", "category": { "id": "/category/commons" }]	Education Film Engineering	
Extract Types in Each Domain	[{ "type": "/type/type", "domain": { "id": "/book" }, "name": null, "id": null }]	Educational Institution Field of study University system	“/Education” is used as an example for domains.
Extract "Instance Count" of Each Type	[{ "type": "/education/educational_institution ", "return": "count" }]	159,711	"estimate-count" should be used for types with very large instance counts, such as “Geocode” and “Film Character”.
Extract Properties of Each Type	[{ "type": "/type/property", "schema": { "id": "/ Education / Educational Institution " }, "id": null, "name": null}]	Campuses School type Number of faculty	The value of a property can also be another topic.
Extract "Included-Type" Relationships	[{ "id": "/education/educational_institution", "/freebase/type_hints/included_types": [{ "id": null }] }]	Topic Organization Employer	“Included Types” should be extracted after the extraction of all types.

Observations on Freebase API Usage

Freebase offers a generous usage capacity for its API usage with a free quota up to 100,000 read calls per day per person or organization [8]. The API read calls were considered as an indicator to the responsiveness of Freebase engine. Table (3) demonstrates the observations on the time taken to complete HTTP read requests, recorded during the extraction process.

TABLE 3. OBSERVATIONS ON FREEBASE RESPONSIVENESS

Procedure	No. of HTTP Requests	Response Time (msec) ≈
Extracting Domains	1	363
Extracting Types	82	6,011
Extracting Instance Counts	2,210	228,949
Extracting Properties	2,210	66,571
Extracting Included Types	2,210	99,941

Storing the Extracted Schema

The extracted schema of Freebase was stored in a relational database. Though Freebase schema is not rigorously relational, it was not difficult to map it into a relational schema. Figure (4) depicts the tables and relationships constructed in the schema.

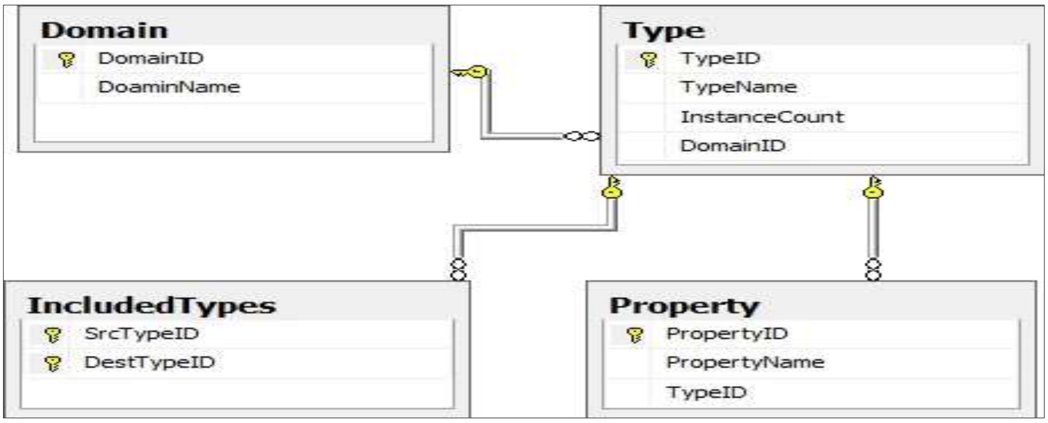

Figure 4. Relational schema for storing the extracted schema.

Constructing the Schema Graph

The extracted schema of Freebase was conceived as a graph, where the graph could be constructed as follows:

1. Nodes: Each node in the schema graph represented one of Freebase types. The total number of the nodes in the graph reached 1,659.
2. Edges: Linking nodes with directed edges was realised by using the “included-type” relationships. For instance, since the “Author” type included the “Person” type, so a directed edge was constructed denoting “Author” as the source node, and “Person” as the destination

node. The total number of directed edges was 2,837. Figure (5) depicts an example of the included-type relationship.

3.

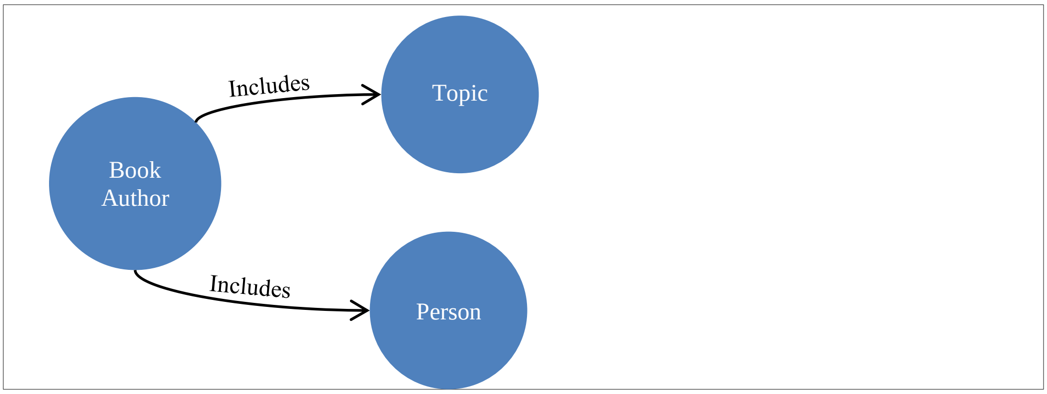

Figure 5. An Example of how nodes were linked in the schema graph through directed edges that represent the included-type relationships.

4. Assignment of Edge Weights: “Instance Count”, a property of Freebase types, was considered to assign weights to edge. The edge weight can determine how influential the source type is with respect to the destination included type. The edge weight was defined as follows:
Edge Weight (W) = $(IC_{\text{Source Type}} / IC_{\text{Included Type}})$
, where
$IC_{\text{Source Type}}$ →Instance Count of the Source Type (Source Node)
$IC_{\text{Included Type}}$ →Instance Count of the Included Type (Destination Node)

Analysis and Visualisation of the Schema Graph

This section presents the graph-based analysis and visualisations of the schema graph. The graph-based analysis of the schema graph aimed at identifying the most influential nodes, which substantially refer to Freebase types. In-degree and Eigenvector centrality were adopted to measure the influence of graph nodes. Furthermore, the schema graph was visualised with emphasis on the most influential nodes. The graph analysis and visualisations were conducted using Gephi [12], an open-source software for network analysis and visualisation.

Identifying High In-Degree Nodes

The node in-degree can interpret how intensively a type is used as an included-type in the schema. Figure (6) demonstrates the top 10 ranked Freebase types by the in-degree measure. On the other hand, the visualisation presented in figure (7) emphasises the nodes with significant high in-degree.

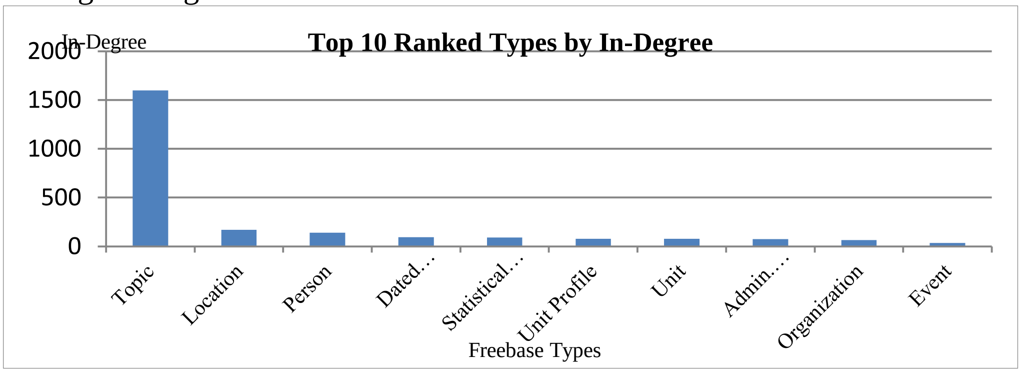

Figure 6. Top 10 ranked Freebase types by in-degree.

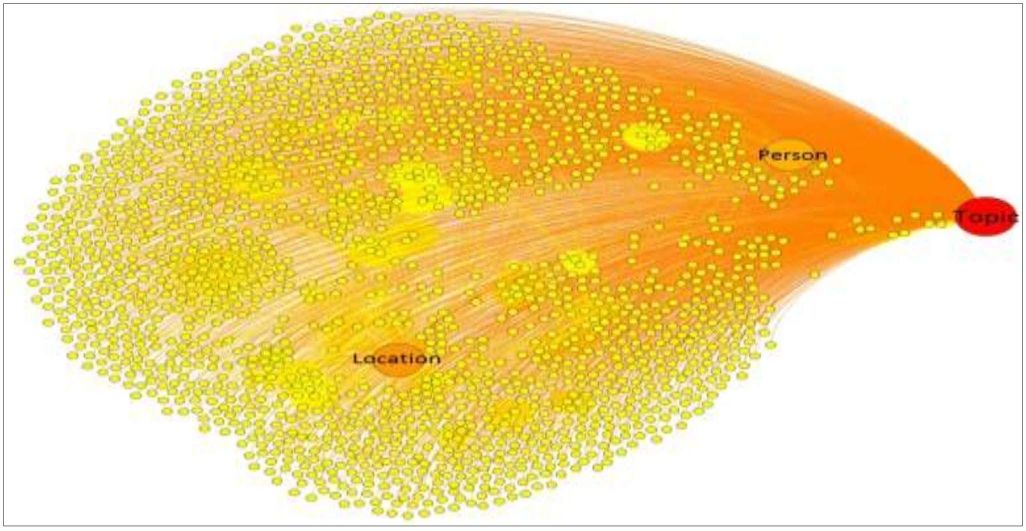

Fig. 7. Freebase schema graph with emphasis on significantly high ranked in-degree nodes. The rank of the node in-degree is represented as the node background colour varying from yellow (lower in-degree) to red (higher in-degree).

Identifying High Centrality Nodes

Eigenvector centrality was used to measure the centrality of nodes since nodes with high degree of centrality can be considered to be more influential [13]. Figure (8) demonstrates the top 10 ranked Freebase types centrality measure. On the other hand, figure (9) visualises the schema graph with emphasis on nodes with significant high centrality.

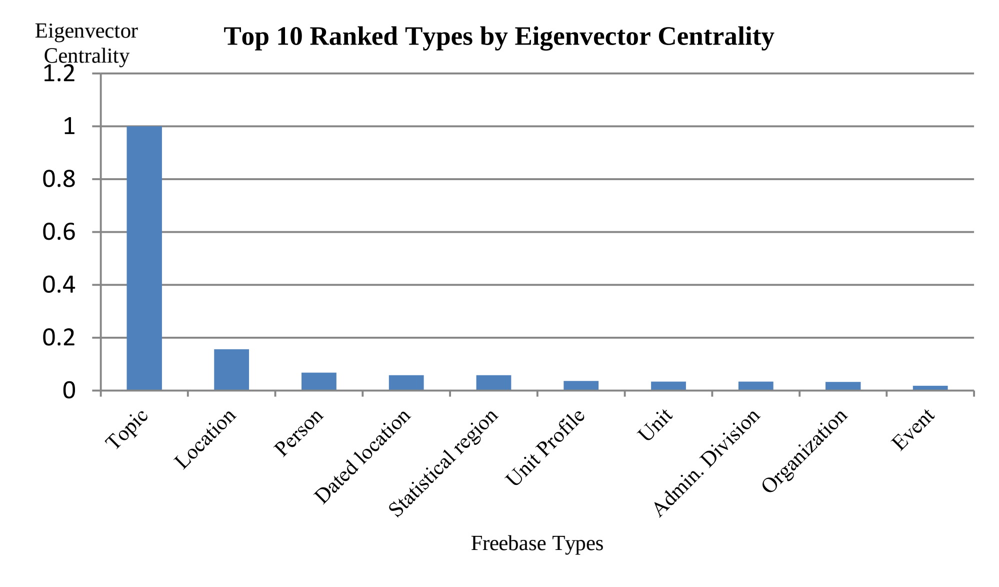

Fig. 8. Top 10 Ranked Types by Eigenvector Centrality.

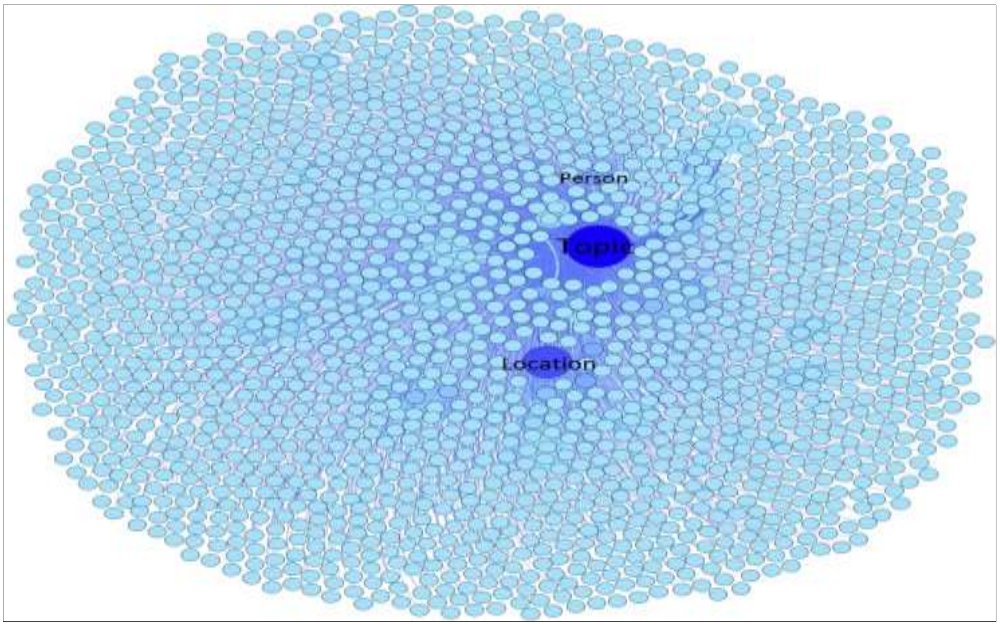

Figure (9): Freebase schema graph with emphasis on significantly high ranked centrality nodes. The rank of the node centrality is represented as the node background colour varying from light blue (lower centrality) to dark blue (higher centrality).

RESULTS

The in-degree and centrality measures were used as indicators to the most influential nodes in the schema graph. "Topic", "Location" and "Person" types were significantly high ranked of both in-degree and Eigenvector centrality measures. The results could be interpreted as that large number of Freebase types should include one or more of those most influential types.

METHODOLOGY LIMITATIONS

The construction of the schema graph depended mainly on two particular properties of Freebase schema, which are "Included Types" and "Instance Count". Therefore, it might not be possible to generalise the adopted approach in order to build other schema graphs unless similar properties are provided. However, the methodology can still be useful with Freebase case for providing graph-based analysis and visualisation purposes.

CONCLUSIONS

In this paper, a graph-driven approach is presented to represent Freebase schema as a directed weighted graph. The approach adopts the included-type relationships to construct directed edges and the instance counts of Freebase types to assign weights to edges.Furthermore, graph-based analysis is conducted to identify the most influential types in Freebase schema using in-degree and Eigenvector centrality measures. The schema graph of Freebase is visualised with emphasis on the in-degree and centrality measures. In essence, the paper embraces an approach of graph-based analysis and visualisation that can help with describing schema structure and summarising the data structure of large-scale schemas, such as Freebase.

REFERENCES

[1] Kurt Bollacker, Robert Cook, Patrick Tufts, “Freebase: A Shared Database of Structured General Human Knowledge”, AAAI'07 Proceedings of the 22nd national conference on Artificial intelligence - Volume 2, 2007.

[2] “http://wiki.freebase.com/wiki/Data_sources”, retrieved on 14/09/2014.

[3] Kurt Bollacker, Patrick Tufts, Tomi Pierce, Robert Cook, “A Platform for Scalable, Collaborative, Structured Information Integration”, Sixth International Workshop on Information Integration on the Web, AAAI, 2007.

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, Jamie Taylor, “Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge”, Proceedings of the ACM SIGMOD international conference on Management of data, 2008.

[5] Hirsch, Christian, John C. Grundy, John G. Hosking. "Thinkbase: A Visual Semantic Wiki." International Semantic Web Conference, 2008.

[6] Hirsch, Christian, John Hosking, John Grundy. "Interactive visualization tools for exploring the semantic graph of large knowledge spaces", Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW2009), vol. 443, 2009.

[7] Mahmoud Elbattah, Mohamed Roshdy, Mostafa Aref, and Abdel-Badeh M. Salem, “Exploring Freebase Potentials for Big Data Challenges",International Journal of Computer and Information Technology (IJCIT) Volume 03, Issue 03, 2014.

[8] https://developers.google.com/freebase/usage-limits, retrieved on 14/09/2014.

[9] http://wiki.freebase.com/wiki/Search, retrieved on 13/09/2014.

[10] http://james.newtonking.com/json, retrieved on 13/09/2014.

[11] http://wiki.freebase.com/wiki/Commons, retrieved on 14/09/2014.

[12] https://gephi.github.io/

[13] M.E.J Newman, “Networks: An Introduction”, Oxford University Press, page 168, 2010.

[14] Tu, Ying, and Han-Wei Shen. "GraphCharter: Combining browsing with query to explore large semantic graphs." Visualization Symposium (PacificVis), 2013 IEEE Pacific. IEEE, 2013.