Evaluating The Semantic Mapping

Dewi Wardani
Department of Informatics
Universitas Sebelas Maret
Surakarta, Indonesia
dww_ok@uns.ac.id

***Abstract*—The increasing of the importance of links in the network of data influences the idea that links should be considered more in the mapping relational to graph model. Semantic abstraction gaps often occur during the mapping process where the link in the real world is mapped as a node in a graph model. This paper focused on evaluating the result of mapping and converting without losing the semantics. We propose the evaluation of our approach by using schema.org as the semantic standard. The experiments in three data sets show that the semantic mapping approach is pretty effective. We obtain quite good score matching without considering the gap index (the average is 0.6922) and with considering the gap index (the average is 0.5264) and the average precision score, 0.7042, is pretty good too.**

Keywords—Semantic Mapping, relational model, graph model, schema.org

I. Introduction

The idea of the inventor of Semantic web is to create the web of data [1]. Sometimes, it is called the network of thing [2]. The data will be equal to a thing than a string or the other data types. The main idea is to make data more connected, meaningful and understandable on the machine side. For this purpose, we need the conceptualization of thing like in the real world. In both theory and practice, preparing data which meet the need of semantic web is the bottom block in this technology architecture. As the connected data, we don't see them as a stand-alone data but as a subgraph. We can see that the data model of the semantic web is a graph-based data model [3] [4]. Graph model itself has been proven useful to help solving some tasks [5] [6]. In this model, we will have not only a network of a node but also a network of a link. Links will be more important to the network of data, and link mining will be important in the near future [7].

Lots of data exist in many domains such as a structured and an unstructured type, yet the model is not semantic web friendly. Especially, the structured data in the relational model, it has huge and broad-area user and has been used successfully for a long time [8]. It also has a high-quality information in the sense that it has the main information in almost all applications. This type of data cannot be ignored as a data source for the semantic web. Concerning its importance, we need to map the relational model to a graph model first, which semantic web friendly and much expressive to represent knowledge in the real world [9]. Later on, it can be mapped to the other model or language such as RDF or OWL. Direct mapping to RDF or OWL is out of the focus of this work.

The mapping relational to graph model usually is a naïve converting process, which put all tuples in each relation as a node and the foreign key as a link between two nodes, but not few scientific papers are written for this approach. Almost all ended their work until a relational model is converted to a graph model, without considering the semantics of the real world. Although a graph model basically has no schema, but the mapping process should be started from the schema, and the rest populated data will be following the schema. In summary, up to our knowledge, the semantic mapping which has been proposed [10] is the first work which maps with considering the higher abstraction of the relational model and has shown good performances comparing the naïve model. Showing good performances is not enough to acknowledge the model keeps up the semantic of the real world. Measuring its effectiveness in the sense not losing the semantics of model is an important issue. On the other side, there is no work which measures whether the result graph model keeps up the semantic of the real world.

The main goal of this work is **to evaluate the effectiveness of the semantic mapping approach by comparing the result with the vocabulary in the *schema.org (http://schema.org)*, a general ontology created by experts as the standard evaluation score**. The remaining sections are organized as follows: section two explains the related work. Section three summarizes the mapping approach from the previous work [10]. Section four describes the evaluating method to examine whether the graph keeps up the semantic of the real world. We conduct and discuss the result of experiments in section five. Finally, the conclusion and discussion our the near future work in section six.

II. Related Work

In a previous work [11], the goal of the authors is to improve the performance in querying the graph model data. The approach aggregates the populated data into one node as far as it's possible in all possibility search queries to reduce the query traversal time. In our point of view, aggregating the data into one node might cause the semantically lose, the node does not have specific semantics. The node is only a bag for holding data. The other work [12], transforms the relational model to graph model based on dependencies between each attribute of the relation. It is similar to the earlier approach which the semantics of relation will be lost, and there is a possibility of graph data redundancy. In general, we think that the mapping process should not only focus on consuming the data, but also can support the mining of the link or other extended works that possibly only can work better by using the graph model data.

Some other works, convert the relational model to RDF or OWL [13] [14] [15] [16] [17] [18], but almost all focus on direct converting without considering the higher abstraction of a relational database. This is what we mean that the converting process has a potential to semantically lose. In general, the earlier works focus on mapping the data, attributes and relation, but still missed in considering the scheme which is the representation of semantic abstraction of the real world.

The network of data can have multiple types of links and complex dependency structures. This fact is a motivation of our work, that mapping relational to graph should keep up the possibility of the heterogeneous network. In the naïve mapping process to graph model, usually the Foreign Key *(FK)* is mapped as a link, but this rule is not enough for keeping the semantics of a heterogeneous network. In this work, we exploit deeper the link structure to be used in our mapping approach without semantically lose and keep up the graph model like in the real world.

III. Link Structure in Relational Model

We proposed the semantic mapping which to our knowledge is the first work mapping relational model to graph model which consider the semantic of links. Previous work ignored the semantic issue of the link that has more than two nodes as the link as well. In graph point of view, it is normal that a link connects to more than two nodes. In the real world is also very common that a relationship relates to more than two things. In this part we summarize some definitions of our previous work [10] as below:

DEFINITION 1 (The Relational Model). Let $S\ (R_1, R_2 \dots R_i)$ be a relational model which consists of a set of Relations, i is the degree of relation, a set of Primary Key $PK\ (PK_1, PK_2 \dots PK_{iPK})$, a set of foreign key $FK\ (FK_1, FK_2 \dots Fk_{iFK})$ and others sets of integrity constrains (*IC*). Each Relation consists of a set of attributes, $R\ (A_1, A_2 \dots A_j)$, j is a degree of attributes. r is a relation instance of R, a set of tuples $r\ (r_1, r_2 \dots r_i)$. Each *n-tuple*, t is an ordered list of r values, $t=(t_1, t_2 \dots t_n)$, each value v_n is a value of attribute A_j.

DEFINITION 2 (The Link Structure). Let $FK_{i\ ->\ j}$ be a foreign key which connects relation R_i to relation R_j. m is the number of in-link and n is the number of out-link where $m \geq 0$, $n \geq 0$.

A number of in-links of Relation R_i as m_i and a number of out-links of Relation R_i as n_i, within assumption there is no cycle relationship.

$$m_i = \sum n_j \quad \text{(i)} \qquad\qquad n_i = \sum m_j \quad \text{(ii)}$$

DEFINITION 3 (The Relation Type Set). Let L be a set of link structure information of relation $L=L_1, L_2 \dots L_i$, each L_i is a list of value $L_i=(m_i, n_i)$ where m_i is the number of in-link of R_i, n_i is the number of out-link of R_i and let *RSink*, *RSource* and *RHub* are a list of a relation R_i which satisfies L data. If in the relation R_i, $n_i=0$ then R_i is a *RSink*. If in the relation R_i, $m_i=0$ then R_i is a *RSource*, otherwise the relation R_i is a *RHub*. Each relation type set (*RSink, RSource, RHub*) is a set of Relation R_i

DEFINITION 4 (The graph model). Let G be a graph model of a relational model S, is a set of node N and a set of edge E, a directed property graph $G(N,E)$ where N is a set of node $N=(N_1, N_2 \dots N_l)$ and $N \epsilon (RHubRSink)$. Each N_i is a set of attribute and its value $N_i = (A_i, V_j)$. The edge is defined as (i) $(\overrightarrow{N_k, N_l}) \in E$ if there is a fk between two relations $R_l - \overrightarrow{fk} - R_k$. The label of the edge is a combination of the label of two relations $R_l R_k$. (ii)* $\overrightarrow{N_r} \in E$ where $R_r - \overrightarrow{fk} - (R_l, R_k)$ and $R_r \in (RSource)$ hence the form of the subgraph is $N_l^A - \overrightarrow{N_r} - (N_k^H)$. Each edge N_r has a set of attribute and its value $N_r = (A_j, V_j)$. The label of the edge is the combination of the name of the relation and the name of hub relation $N_r N_k^H$. (iii) $(N_r) \in HE$ where $HE\, is\, HyperEdge$ and where $R_r - \overrightarrow{fk} - (R_1, R_2 \dots R_i)$ and $R_r \in (RSource)$ hence the form of the subgraph is $N_r - \overrightarrow{HE} - (N_1, N_2 \dots N_i)$. N_r has a set of attribute and its value $N_r = (A_j, V_j)$. The label of the hyperedge is the name of the relation N_r.

*) N_l^A is the node as an Authority and (N_k^H) is the node as a Hub. The weight of Authority A_i means how famous node N_i is referenced by the other nodes. The weight of Hub H_i means how often node N_i refers to the other nodes. Between two nodes N_l and N_k we calculate which node has the greater weight of authority $\nabla A_i = A_i - H_i$. N_l^A draws the direction of an edge. Here are the formulas to find the type of node whether as an authority node or a hub node.

$$A_{Ri} = \sum outlink_{Rj} \quad \text{(iii)} \qquad\qquad H_{Ri} = \sum inlink_{Rj} \quad \text{(iv)}$$

g is a graph data, $g=(n, e)$ which following $G=(N, E)$ as a graph model.

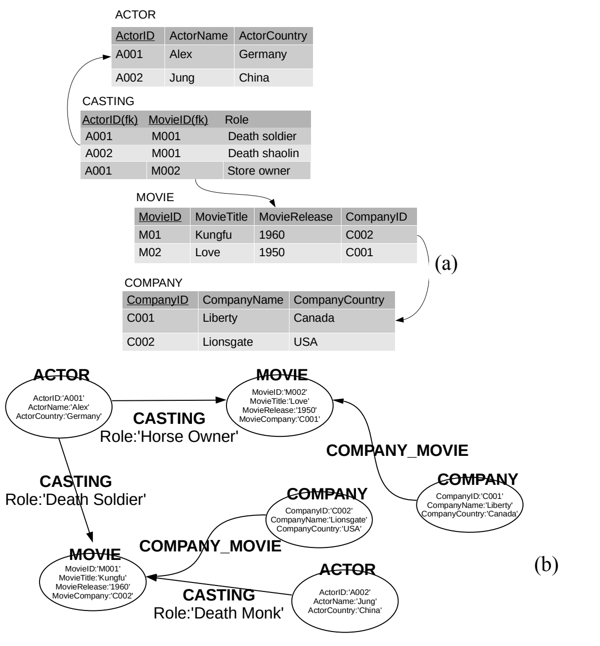

Fig 1. The example mapping and converting (a) relational model to (b). a property graph model based on the approach of the previous work [10]

Identify applicable funding agency here. If none, delete this text box.

Figure 1 shows that the approach mapping returns a relational model to a property graph. Figure 1 is the result example of the Semantic Mapping. We can see that RSink is relation Actor and Company. RHub is relation Movie and RSource is relation Casting. Therefore, Casting becomes a link and the others become nodes. Based on our above definitions, the direction of all links are provided as well and is drawn in Fig 1.b.

IV. EVALUATING THE SEMANTIC MAPPING

To measure the effectiveness the mapping approach which is explained in section 3, we will compare the result of our graph model with vocabulary and conceptualization of the real world which is formed in the ontology. The measuring process basically is a matching process between our graph model with a graph model of ontology. We use *schema.org*, a famous general vocabulary ontology. We define the graph model in Ontology, in DEFINITION 5 (for the sake of easy reading, we continue the definition and formula numbering from the section 3).

DEFINITION 5 (The graph of ontology). Let O be an ontology which consists a set of classes $C_1, C_2 ... C_n$ in hierarchy format. It has an index of the cluster of class i where $i=0,1,... j$. Each class C_n can have another set of subclasses $C_1, C_2 ... C_m$ in each index. Class (domain) C_D is connected to another class (range) C_R through object property OP_a or data property DP_b . Hence, a subgraph of ontology is $C_D \circ \vec{P}_x \circ C_R$ where $P_x \in (\vec{OP}_a, \vec{DP}_b)$.

The special case in *schema.org*, it has class ACTION which especially describes what entities do. In reality, the class is a relationship therefore it should be as a property. The role class domain and object property are reversed. Hence, the subgraph under the class ACTION is $P_x \circ \vec{C}_D \circ C_R$

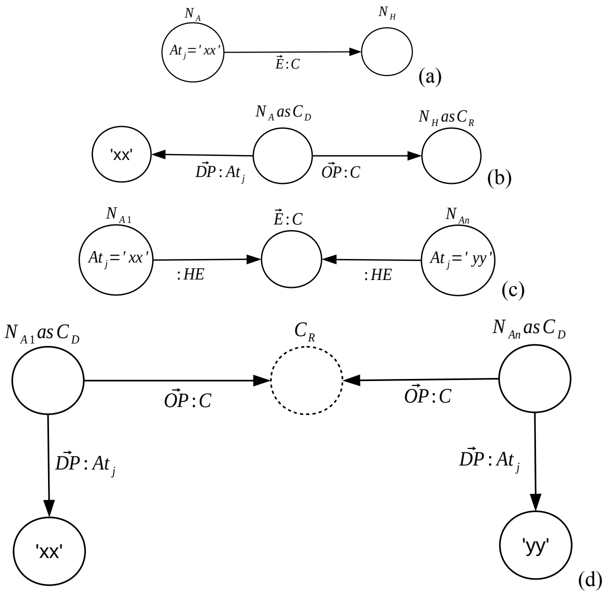

Fig 2. A subgraph type 1 and 2 (a) in our graph model (b) when it's formed in a subgraph of ontology. A subgraph type 3 (c) in our graph model and (d) when it's formed in a subgraph of ontology *n* is a number of the authority node N_A.

We will measure whether the links $\vec{E}$ which are obtained by our approach match with the properties $\vec{P}$ in ontology. To measure it, we should concern also the nodes which are connected by the link. The link can be varied in the heterogeneous network. The same or similar link does not always connect to the same nodes. Hence, we will match each path of link $N_A \circ \vec{E} \circ N_H$ in our graph model with path subgraph of property $C_D \circ \vec{P}_x \circ C_R$ in the ontology. In conceptual point of view, actually DP_b is more matched with attribute At_j in node N_i , but in this work we follow formalization of ontology that only matching process $\vec{E}$ to $\vec{P}$ not At_j to $\vec{P}$. The best score of similarity matching will be obtained if the path $N_A \circ \vec{E}$ is matched with $C_D \circ \vec{P}$ in the same index cluster of a class of ontology. Therefore, we use d as a distance factor to calculate the class gap between the matching result of $S(N_A, C_D)$ and $S(\vec{E}, \vec{P})$. Considering that C_R usually cross the hierarchy of ontology then we avoid similarity $S(N_H, C_R)$ and avoid the distance factor between $\vec{E}$ and N_H .

V. EXPERIMENT AND RESULT

We conducted experiments for three data sets, (1) The artificial movie database, (2) The example problem in [11], to compare the difference result with our approach and (3) The real data set of IMDB (*http://www.imdb.com/*). The measuring process focused on measuring link in the graph result. We implemented two ways experiment: experiments with and without considering the *gapindex* between the link and the node within class and property network of ontology. We use the famous matching score, cosine similarity to calculate the content similarity. The same syntax might have two different meanings or different syntax might have the same meaning, therefore, we use WordNet (*http://wordnet.princeton.edu/*) similarity too. We also calculate the precision score to measure the relevance of the experimental result. In all experiments, we differentiate score between with (with d) and without considering the distance measure and

A. The Artificial Movie Database

It has 10 relations, *ACTOR, AWARD_WINNER, CASTING, CATEGORY_WINNER, COMPANY, DIRECTOR, LOCATION, MOVIE, MOVIE_FESTIVAL* and *PRODUCTION.*

By using our approach 3 relations *(CASTING, AWARD_WINNER, PRODUCTION)* turn out as links and the rest 7 relations turn out as nodes. *CASTING_ACTOR* is a link type2. Whereas *AWARD_WINNER* and *PRODUCTION* are a link type3. In the real world, it's also correct that *CASTING, AWARD_WINNER* (win specific festival) and *PRODUCTION* (the process of movie making) are relationships and are not entities. The others, *MOVIE, COMPANY, CATEGORY_WINNER, DIRECTOR, ACTOR, and LOCATION* are entities. A link *COMPANY_MOVIE* is a link type1, which is the only one real link within relational model.

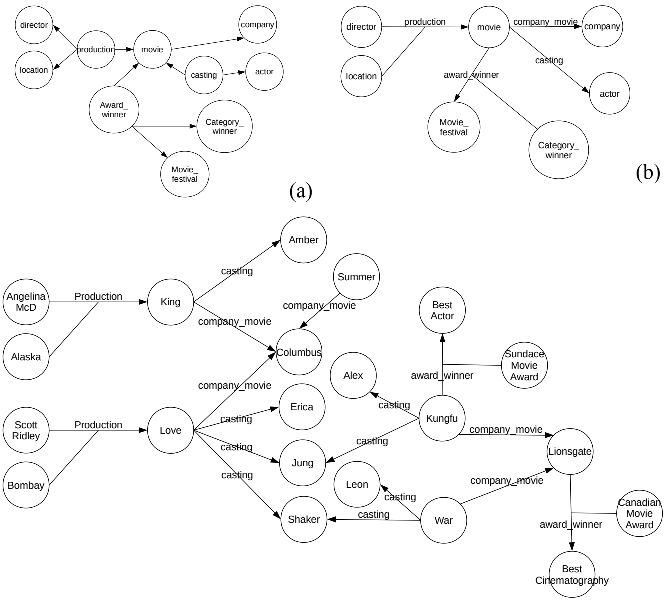

Fig 3. (a) The graph model of naïve approach, (b) The graph model of our approach, (c) The mapping result of graph data

TABLE I. THE MATCHING SCORES OF THE ARTIFICIAL MOVIE DATABASE

Link	WordNet Matching Score		Cosine Matching Score	
	without d	with d	without d	with d
COMPANY_MOVIE	0.6861	0,3430	0.6034	0,3017
CASTING_ACTOR	0.7380	0.7380	0.8535	0.8535
AWARD_WINNER	0.8150	0.8150	0.7831	0.7831
PRODUCTION	0.8218	0.4109	0.7628	0.3814

On average *gapindex* is 1. For an example in the link $\vec{E}$ *COMPANY_MOVIE*, which the authority node N_A is *COMPANY*, the domain class C_D *COMPANY* is located upper 1 index hierarchy from the domain class of the relevant property $\vec{P}$, but in the same cluster. We can describe this situation is that the N_A match with subclass-of C but the link $\vec{E}$ is connected to the class C . In the link *CASTING_ACTOR*, there is no *gapindex* , the authority node N_A *MOVIE* is in the same class cluster. It means that the path $N_A \circ \vec{E}$ match with path $C_D \circ \vec{P}$ within one hierarchy class cluster.

We also calculate the precision score, we obtained on average is *0.6483* for WordNet similarity and on average is *0.7739* for cosine similarity. The matching score will be decreased but the relevance of a link $\vec{E}$ in our graph model with a property $\vec{P}$ of ontology is maintained stable. This situation often occurs, also in the other 2 data sets. Hence, the precision score is still pretty good, even though the matching score is decreased. It indicates that the link which is obtained by the semantic mapping approach is valid enough. The result from both WordNet similarity and cosine similarity are pretty good and also good for the precision score. It indicates that the approach is promising.

B. The Social Database

It has 5 relations, *USER, FOLLOWER, TAG, BLOG* and *COMMENT.*

TABLE II. THE MATCHING SCORES GRAPH MODEL OF THE SOCIAL DATABASE

Link	WordNet Matching Score		Cosine Matching Score	
	without d	with d	without d	with d
TAG_COMMENT	0.7205	0.3602	0.7071	0,3535
FOLLOWER_BLOG	0.8705	0.8705	0.6034	0.6034

The precision score on the average is *0.7916* for WordNet similarity and on average is *0.5833* for cosine similarity. 2 relations *(FOLLOWER, TAG)* turn out as links and the rest 3 relations turn out as nodes. Both are links type2. In the real world, it's also acceptable that, *FOLLOWER* (action to follow) and *TAG* (commenting action) are a relationship between entities and not the entity. The others, USER, *COMMENT,* and *BLOG* are entities. The matching score and precision score are all pretty good.

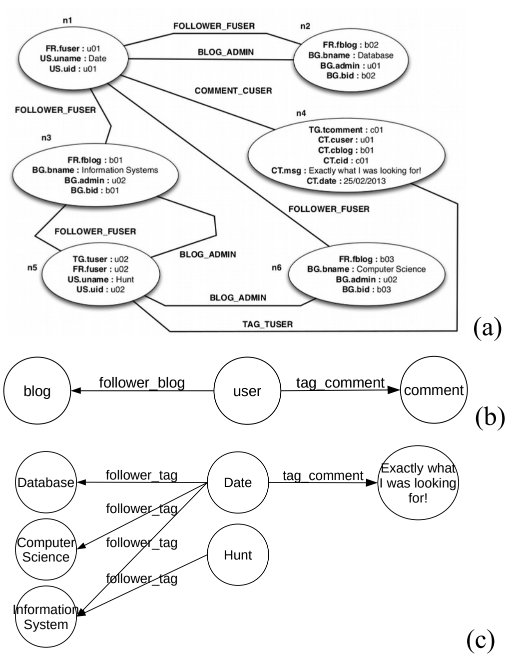

Fig 4. (a) The graph model of [11] , (b) The graph model of our approach and (c) The mapping result of graph data

C. The IMDB Data Set

It is the real movie data set. It has 21 relations, *CHAR_NAME, COMPANY-NAME, COMPANY_TYPE, COMPLETE_CAST_TYPE, INFO_TYPE, KEYWORD, KIND_TYPE, LINK_TYPE, NAME, ROLE_TYPE,*

AKA_NAME, TITLE, CAST_INFO, MOVIE_COMPANIES, AKA_TITLE, COMPLETE_CAST, MOVIE_INFO, MOVIE_INFO_IDX, MOVIE_KEYWORD, MOVIE_LINK and PERSON_INFO.

TABLE III. THE MATCHING SCORES GRAPH MODEL OF THE SOCIAL DATABASE

Link	WordNet Matching Score		Cosine Matching Score	
	without d	with d	without d	with d
NAME_AKA_NAME	0.6333	0.3166	0.3535	0.1767
CAST_INFO	0.6461	0.6461	0.4259	0.4259
MOVIE_COMPANIES	0.5671	0.5671	0.6025	0.6025
COMPLETE_CAST_COMP_CAST_TYPE	0.3809	0.3809	0.7041	0.2347
MOVIE_INFO_INFO_TYPE	0.5512	0.5512	0.6666	0.3333
MOVIE_INFO_IDX_INFO_TYPE	0.5512	0.5512	0.6666	0.3333
MOVIE_KEYWORD_KEYWORD	0.6666	0.6666	0.8535	0.8535
MOVIE_LINK_LINK_TYPE	0.3430	0.1715	0.7041	0.3520
PERSON_INFO_INFO_TYPE	0.8056	0.2685	0.2499	0.1249
AKA_TITLE_KIND_TYPE	0.6428	0.6428	0.8535	0.4267

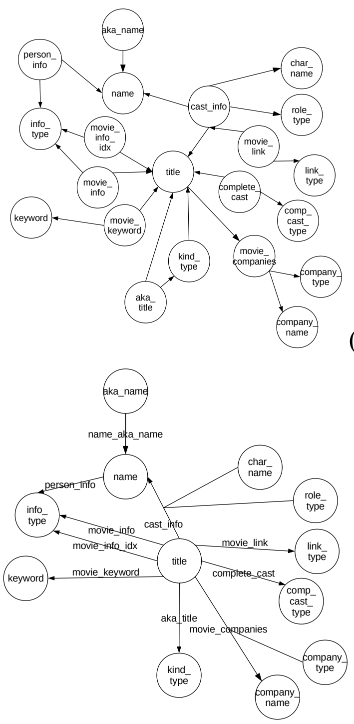

Fig 5. (a) The graph model of naïve approach, (b) The graph model of our approach

The precision score on the average is *0.6726* for WordNet similarity and on average is *0.5700* for cosine similarity. 8 relations *(CAST_INFO, MOVIE_COMPANIES, COMPLETE_CAST, MOVIE_INFO, MOVIE_INFO_IDX, MOVIE_KEYWORD, MOVIE_LINK, PERSON_INFO)* turn out as links and the rest 14 relations turn out as nodes. The matching score and precision score are all pretty good. A few links have low scores. We notice that the low score is caused by a gap of syntax between terms in graph model and the ontology. For an example, in the second experiment for The Social data set, we have a candidate relation has a label *TAG*. *TAG* has information user and their comments. The same description is defined as *COMMENT* or *REVIEW* in the ontology. The same case in the third data set of The IMDB data set, the candidate relation has a label *MOVIE_LINK_LINK_TYPE* and it is a link in IMDB dataset. The same description is defined as URL in the ontology. Those terms refer to the same meaning but as we can see, those terms have a syntax gap.

In the small data set, it might be easier to map manually, but it could be hard and tiring for huge and complex relational model. Although the matching score on the average is not really high but still pretty good, also the precision score is quite high. Therefore, the approach is effective enough to map relational model to graph model without semantically lose than manual semantic mapping especially for a very complex relational schema which probably has hundreds of relation. From the real IMDB data set, we learn that in the complex and huge relational schema there is a bigger possibility many relations which should be mapped as a link, not as a node. Even though the data set of the experiment are the result of this approach [10], but basically all obtained graph models from the other approaches can be evaluated as well.

VI. CONCLUSION AND FUTURE WORK.

We have proposed the measure to evaluate the semantic mapping, a new approach in mapping and converting relational database model to graph model. The goal of the semantic mapping is to avoid semantics lose by considering semantic abstraction of the real world. This approach exploits the link structure which naturally lies in a relational model and formulate in a few definitions. The experimental result of three data sets included real data set from IMDB data show that this approach is promising, although the matching scores are not really high. Even though some scores are low, but they are investigated correctly as the concept, in the sense links are mapped as properties in comparing ontology. The average of matching scores is similar each other. The average of the matching score without considering the gap of the index is *0.6922*, and with considering the gap of the index is *0.5264.* On average, the precision score is *0.6796.* The average of WordNet similarity is *0.6517*, a slightly higher than *0.5674* as the average score of cosine similarity. We notice that a gap of the syntax of terms in graph model and ontology causes the low scores.

In the near future, we are going to deploy the graph model which is based on RDF. As both are graph it would be possible to carry out this idea. The most interesting idea is that there is a possibility to introduce a new model of knowledge representation and its implementation. The result of our mapping is a heterogeneous network which really closes with the real world. We will still exploit the heterogeneous link structure of our mapping result in a few extended works such as using it to improve the inference process and working in retrieval based on link structure.

References

[1] T. Berners-Lee and M. Fischetti, *Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor*. Harper San Francisco, 1999.

[2] T. Berners-Lee, J. Hendler, O. Lassila, and others, “The semantic web,” *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[3] C. Bizer, T. Heath, and T. Berners-Lee, “Linked data-the story so far,” *Int. J. Semantic Web Inf. Syst.*, vol. 5, no. 3, pp. 1–22, 2009.

[4] C. Bizer, T. Heath, D. Ayers, and Y. Raimond, *Interlinking open data on the web*.

[5] W. Chen, L. Cao, X. Chen, and K. Huang, “An equalized global graph model-based approach for multicamera object tracking,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 11, pp. 2367–2381, 2017.

[6] S. Safaei, P. J. Blanco, L. O. Müller, L. R. Hellevik, and P. J. Hunter, “Bond Graph Model of Cerebral Circulation: Toward Clinically Feasible Systemic Blood Flow Simulations,” Frontiers in physiology, vol. 9, p. 148, 2018.

[7] L. Getoor and C. P. Diehl, “Link Mining: A Survey,” *SIGKDD Explor. Newsl.*, vol. 7, no. 2, pp. 3–12, Dec. 2005.

[8] M. J. Cafarella, A. Halevy, and J. Madhavan, “Structured data on the web,” *Communications of the ACM*, vol. 54, no. 2, pp. 72–79, 2011.

[9] C. C. Aggarwal and H. Wang, *Managing and mining graph data*, vol. 40. Springer, 2010.

[10] D. W. Wardani and J. Küng, “Semantic Mapping Relational to a Directed Property Hypergraph Model,” in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, 2015, pp. 152–159.

[11] R. De Virgilio, A. Maccioni, and R. Torlone, “Converting relational to graph databases,” in *First International Workshop on Graph Data Management Experiences and Systems*, New York, New York, 2013, pp. 1:1–1:6.

[12] S. Bordoloi and B. Kalita, “Designing Graph Database Models From Existing Relational Databases,” *International Journal of Computer Applications (0975–8887) Volume*, vol. 74, pp. 25–31, 2013.

[13] P. T. T. Thuy, N. D. Thuan, Y. Han, K. Park, and Y.-K. Lee, “RDB2RDF: completed transformation from relational database into RDF ontology,” in *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, 2014, p. 88.

[14] C. Bizer, “D2R MAP - A Database to RDF Mapping Language,” in *World Wide Web Conference Series*, 2003.

[15] S. Rikacovs and J. Barzdins, “Export of Relational Databases to RDF Databases: A Case Study,” in *Perspectives in Business Informatics Research*, Springer, 2010, pp. 203–211.

[16] S. Zhou, “Exposing relational database as RDF,” in *Industrial and Information Systems (IIS), 2010 2nd International Conference on*, 2010, vol. 2, pp. 237–240.

[17] G. Būmans and K. Čerāns, “RDB2OWL: a practical approach for transforming RDB data into RDF/OWL,” in *Proceedings of the 6th International Conference on Semantic Systems*, 2010, p. 25.

[18] N. Konstantinou and D.-E. Spanos, “Creating Linked Data from Relational Databases,” in *Materializing the Web of Linked Data*, Springer International Publishing, 2015, pp. 73–102.