

Semantic Schema Matching Using DBpedia

Saira Gillani

Centre of Research in Networks & Telecom, M. A. Jinnah University Islamabad, Pakistan
Email: sairagilani@yahoo.com

Muhammad Naeem

Department of Computer Science, M. A. Jinnah University, Islamabad, Pakistan
Email: naeems.naeem@gmail.com

Raja Habibullah

Department of Computer Science, M. A. Jinnah University, Islamabad, Pakistan
Email: rajahbib@gmail.com

Amir Qayyum

Centre of Research in Networks & Telecom, M. A. Jinnah University, Islamabad, Pakistan
Email: aqayyum@ieee.org

Abstract— In semantic computing, *Match* is an operator that takes as an input two graph-like structures; it can be database schemas or XML schemas and generates a mapping between the corresponding nodes of the two graphs. In semantic schema matching, we attempt to explore the mappings between the two schemas; based on their semantics by employing any semantic similarity measure. In this study, we have defined taxonomy of all possible semantic similarity measures; moreover we also proposed an approach that exploits semantic relations stored in the DBpedia dataset while utilizing a hybrid ranking system to dig out the similarity between nodes of the two graphs.

Index Terms— Data Component; Schema; Similarity Measure; DBpedia

I. Introduction

There are two levels of data definition; one level defines physical structure of data and the other is characterized by logical level commonly known as schema. A schema implies a plan. A relation schema is the logical definition of an entity that defines the entity name and its attributes with data types. The collection of these relational schemas is called database. Database schema means a Structure of a database that describes how its concepts, their relationships and constraints are arranged. The application of database schema is useful when there is a requirement to integrate different application specified databases. Concepts of various database schemas are defined according to specific domain and requirements at a particular time. Therefore, they possess strong differences from each other. This arise heterogeneity as a highlighted issue. This

heterogeneity may be structural or semantic. Structural heterogeneity includes conflicts like type conflicts, dependency conflicts, key conflicts, behavioral conflicts [1] or semantic conflicts, the differences among the databases that are related to the meaning, interpretation, and intended use of data [2]. To overcome this heterogeneity problem, schema matching is performed. Schema matching is a process in which semantic correspondences are identified between elements of many database schemas. It takes two schemas as input and returns a set of rules r

$$r=(c, p, s)$$

Where

c represents a correspondence between two elements
 s represents the similarity value and
 p defines predicate.

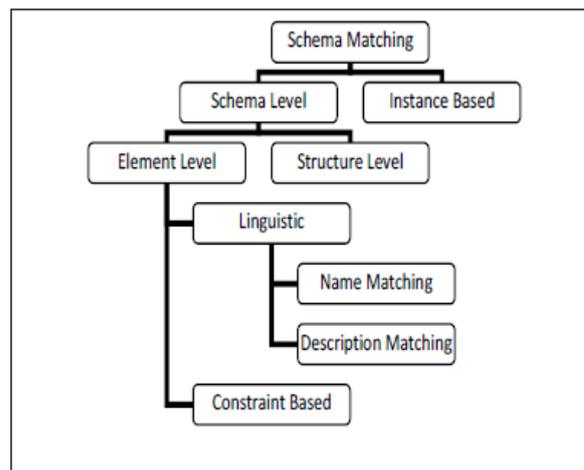


Fig. 1: Example of a figure caption. (figure caption)

There are different database application domains where schema matching is required like integration of web data sources, data warehouse loading, E-business and semantic query processing [3]. The literature describes many schema matching approaches. The application domain defines which matcher would be selected. The following criterion [4] is useful to be considered for individual matchers:

1.1 Instance vs. schema

Matching can be performed on instance data or only on schema-level information.

1.2 Element vs. structure matching

In such matchers, Matching is performed usually on individual elements. In this process, entities are analyzed in isolation for example, attributes while in structure matching; matching is performed on the combinations of elements by analyzing the relations among entities [4-5].

1.3 Language vs. constraint

Linguistic approach is based on names and textual descriptions of schema elements while in constraint-based approach depend on keys and relationships.

1.4 Matching cardinality

The overall match result may relate one or more elements of one schema to one or more elements of the other, yielding four cases: 1:1, 1:n, n:1, n:m.

1.5 Auxiliary information:

In some matchers, auxiliary information, such as dictionaries, global schemas, previous matching decisions, and user input are also involved as input along with schema S1 and schema S2.

1.6 Linguistic Approaches

A main type of schema matching approaches is linguistic approach. In linguistic approaches for matching the schema elements, we use the names and text to discover the similarity between the schema elements. There are two schema level linguistic approaches, name matching and description matching.

1.7 Name Matching Algorithms

The schema elements with similar or equal names are matched. The match can be found in many different ways. The match can be found if the two elements are synonyms e.g. car and automobile. The two elements can also be hypernyms i.e. element from one schema can have a broad meaning and the element from the other schema may fall under it e.g. color is a *hypernym* of red. Common substrings can also be used to find the similarity of names. There are different techniques based on common substrings. These include edit

distance, n-grams and soundex etc. Thesauri and dictionaries are used for the linguistic approaches based on synonyms and hypernyms. General natural language dictionaries, multi-lingual dictionaries and domain-specific dictionaries can be used for this purpose. The name-based approach can find multiple matches for one schema element e.g. "phone" can be matched with both "home phone" and "office phone".

1.8 Description Based Approaches

This approach employs the comments which the schema elements contain [4]. These comments describe the semantics of the elements. The keywords from the description can be extracted and then synonym based approach can be applied on them. E.g. a match can be found between

S1: empn // employee name
and

S2: name // name of employee based on the comments.

In this paper, we have proposed an approach for semantic schema matching using the semantics available in DBpedia dataset. DBpedia has the advantage that it evolves as Wikipedia changes. DBpedia also covers a lot of domains.

DBpedia is basically a machine-understandable equivalent of Wikipedia project. Different queries can be asked against DBpedia using its SPARQL endpoint <http://dbpedia.org/sparql>. The DBpedia knowledge base currently describes more than 3.5 million things. All this information is stored in RDF triples. DBpedia evolves as Wikipedia changes, so there are no issues of domain coverage or content freshness in DBpedia. SPARQL is an RDF query language, which became an official W3C recommendation in 2008. It is used to query the RDF datasets. The results in SPARQL are returned as result sets or as RDF graphs. SPARQL is similar to SQL in some aspects with a trivial example of query where a query in SPARQL such as SQL comprise of SELECT and WHERE clauses.

The remaining of this article is structured as follows: in section 2 we discuss some related technologies. In section 3 we define taxonomy of semantic similarity measures. In section 4, we discuss our proposed system. In section 5, we have discussed some results of our proposed approach. Finally in section 6, we have our finding in conclusion and also give some future work.

II. Related Work

2.1 Existing problem and solution

To predict semantic similarity between different XML schema documents Jeong et al., proposed a novel supervised learning technique (i.e., neural network-based partial least squares, NNPLS) [6]. In this technique, before comparing two XML schemas, the neural

network classifier needs to be trained. This technique consists of two phases: one is offline preparation in which more suitable supervised learner is selected while the other is online matching phase in which matching between similar pair is performed. For this technique, they used different similarity measures, both in lexical and structural.

The Similarity Flooding (SF) [7] approach is based on a fix point computation in which idea of similarity propagation is used. In this scheme, different schemas (SQL DDL, RDF, XML) are converted into directed labeled graphs. SF is used to determine the match between corresponding nodes. If two nodes from two schemas are different then their neighbors may also be similar. Therefore, if the adjacent nodes are similar then two concepts are related. The algorithm relies only on syntactic techniques like string-based comparison of nodes' labels to obtain an initial alignment which is further refined within the fix-point computation. Fix-point computation is used to determine correspondences of 1:1 local and m:n global cardinality between corresponding nodes of the directed labeled graphs. The basic philosophy of SF is based on similarity propagation so similarity spreads from similar nodes to adjacent neighbor nodes and this similarity increases in each iteration until the fix-point is reached. But there are some drawbacks of this scheme like it cannot overcome the cardinality related problems of the direct ancestors and if labels are often identical then this scheme does not show good performance. To find semantic text similarity, Islam *et al.* proposed a technique by using corpus based word similarity and string similarity. Their technique consists of two methods [8]; one is to determine semantic similarity of words and the other is to find automatic word segmentation. For matching schemas, they used a single property (element name) by using natural language processing technique. To calculate the relative semantic similarity, they used a longest common subsequence (LCS) [9] with some normalization and Second Order Co-occurrence PMI (SOC-PMI) word similarity method [10].

Duchateau *et al.*, presented an automated method to calculate semantic similarity measure between two schema elements [11]. They also proposed a tool Approxivect which is based on the approximation of terminological methods and on the cosine measure between context vectors. Vectors are composed of neighbor nodes and to find similarity between them they compared with the cosine measure. At the end Approxivect aggregates the results obtained by several similarity measures to improve the quality of discovered matches. This method is not for any specific domain and does not use any type of dictionary or ontology. However, there are some drawbacks of using aggregation function: to apply all similarity measures to all elements is costly in time and resource perspective, each match algorithm has its own distribution value so threshold of every algorithm should be their own while

in this process a threshold is applied on the aggregated value, and aggregation function decreases the flexibility to add new match algorithms. To enhance the performance of matching between XML tree structures, Duchateau *et al.*, used a B-tree index structure [12]. B-tree approach uses indexes which increase the performance of the matching process by reducing the number of matching possibilities. They compared B-tree approach with their own Approxivect approach [11]. Results showed B-tree approach better results in performance than Approxivect.

Authors discussed content based schema matching [13]. For finding similarity they used Entropy Based Distribution (EBD) method which is based on N-gram. To determine similarity their method relies on shared instance data. So if the data having shared instances then their method will give correct results otherwise it will produce incorrect results. To overcome this problem, authors of [14] proposed two new algorithms for schema matching. First they select attributes for comparison from both targeting schemas. Then TSim algorithm extracts distinct keywords from compared attributes and this algorithm is not based on shared instance data. This algorithm uses K-medoid clustering to group all similar themed keywords of same type. For assigning keywords to clusters they used Normalized Google Distance (NGD) as a distance metric. Then they calculate the conditional entropy of each cluster by using the number of occurrences of each keyword in the cluster. And this conditional entropy value of each cluster will be used in the final EBD calculation between the two attributes. They claimed that their approach is free from the syntactic requirements of N-grams.

In this paper [15] an approach called semantic matching is proposed. In this approach, semantic correspondences are found by mapping meanings of the nodes and not just the labels. Firstly, the two graphs are extracted from the DB, XML or OODB schemas. Then the element-level semantic matching is found. The semantic relations are computed among all the concepts. In the third step, the concepts are computed at the nodes. We start from the root node of the graph and we attach to all the nodes the concepts of all the nodes above them. Then we construct a propositional formula, which represents the matching problem. We translate all the semantic relations into propositional. During this translation, subset translates into implication; equality translates into equivalence and disjointness is translated into the negation of conjunction. In the last step, this algorithm is run iteratively. These iterations help in case, no matching is found or a weak matching is found. This step takes the results from the previous iteration and tries to improve the quality of the final result.

Similarly to our proposed semantic approach, the approach proposed by Mirizzi *et al.* [16]. They also use the DBpedia and external information sources to generate the semantically related keywords and to rank them. The difference between their approach is that we

are using the DMOZ [15] directory to get the human perspective, but they used social tagging systems such as Delicious, which are usually non-moderated and so can produced incorrect similarity. DMOZ on the other hand is maintained and edited thoroughly by human editors and so produces better results as compared to social tagging systems. There are also many different algorithms based on PageRank [17]. These algorithms [18-21] are useful for ranking the RDF resources, but they do not take into account the fact that there is a significant difference between ranking web documents and ranking the resources which have semantic attached to them. Page Rank algorithm considers the origin of the links. These links are only the hyperlinks and have the same relevance. But in case of RDF resources, there can exist more than one type of links. And also the RDF resources may have different contexts and origins as well.

III. Semantic Similarity Measures

We can define similarity as increasing function of comparing the features common between the two objects and decreasing function of comparing features distinct between two objects. The main objective of schema matching is to find similarity between two different schemas. For this purpose in literature, different similarity measures are used like element similarity measures, structural similarity measures, lexical similarity measures, logical similarity measures and semantic similarity measures. Every paper defines one or two similarity measures of specific type but there

is no any paper which can describe the taxonomy of semantic similarity measures. In this paper our focus is to define taxonomy of different semantic similarity measures.

Many semantic similarity measures with a variety of interesting properties have been proposed in literature. In what follows, we present taxonomy of measures of semantic similarity with brief description. These types of Semantic similarity measures can be classified as:

3.1 Lexical Semantic Similarity Measures

Lexical Semantic Similarity Measures which find similarity between two Words, how much information the two concepts share. However, lexical semantic similarity measures provide vague recommendations because these measures concentrate only on labels and their definitions not on context of these labels.

3.2 Structural Similarity Metric

These measures use structural semantics. In these measures the main focus is not on the individual labels and definitions rather on the context surrounding these labels. These measures need some other measures to establish initial relationships between individual labels [22].

3.3 Logical similarity measure

This type of measures is model based. It does not depend on labels and grammar. It is a best measure that efficaciously describes semantics of attributes of objects.

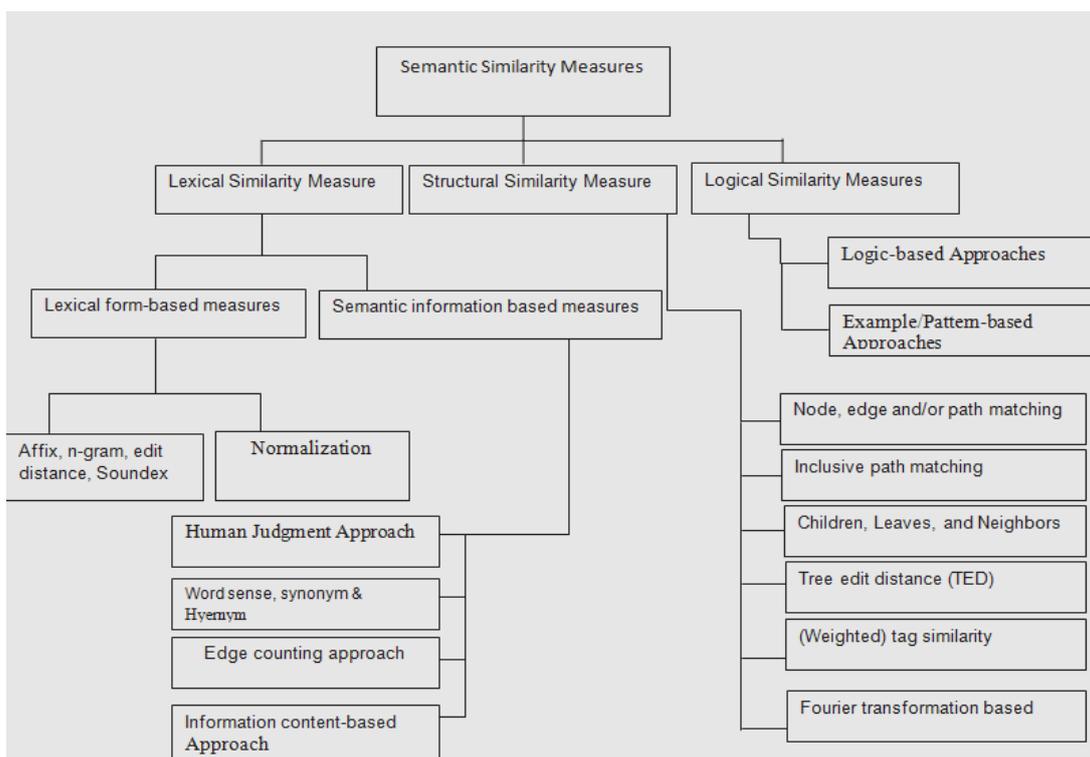


Fig. 2: Taxonomy of Semantic Similarity Measures

IV. Proposed Approach

Our system takes two graphs as an input and finds semantic similarity between them. A node can be represented by a DBpedia resource. The DBpedia resources are URIs, which are unique for each concept. Our system uses semantic relations stored in the DBpedia dataset and uses a hybrid ranking system to find similarity between two resources.

DBpedia [23] is a semantic form of Wikipedia. In DBpedia, the structured information from Wikipedia is

extracted and is made available on the Web as an RDF dataset.

Using DBpedia datasets have some advantages. Compared to other subject hierarchies, in DBpedia, each subject is backed by a rich description including abstracts in many different languages. Another advantage compared to static hierarchies is that DBpedia evolves as Wikipedia changes. So, there are no problems of domain coverage or content freshness in DBpedia. The main components of the architecture proposed are shown in Figure 3.

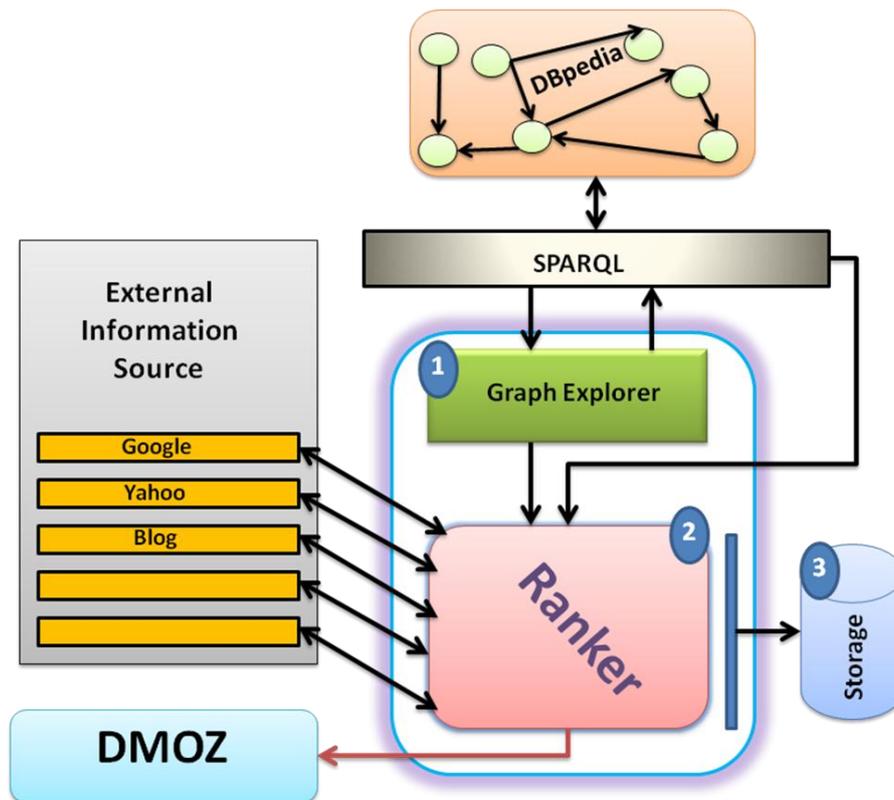


Fig. 3: Architecture of Proposed System

4.1 Graph Explorer

Graph explorer takes a DBpedia url as an input and looks for other urls connected to it via a set of properties. The property used in this approach is `skos:subject`. The `skos:subject` links a resource to its corresponding Wikipedia category.

The url is explored up to a distance of $\text{Max-Depth} = 2$. If the distance is increased from 2, we get many irrelevant resources [16]. Within first two hops, the resources are highly correlated with the root node but when we go to the third hop and beyond, this correlation decreases significantly.

4.2 Hybrid Ranker

The second component of this architecture is a hybrid ranker. The ranker takes two nodes and finds the

similarity between them based on the external information sources such as search engines and DMOZ [24] (Open Directory Project by Google). The purpose of using these different external information sources is to use different perspective. The system uses three different search engines so that it gets the perspective of not only one search engine but all of them. It uses DMOZ to find the human perspective on the similarity of two resources.

It also exploits the Wikipedia related information contained in the DBpedia. It takes two DBpedia resources `uri1` and `uri2` and verifies how many web pages contain or are tagged by the value of `rdfs:label` of the two uris. It then compares it with the number of pages containing (or are tagged by) both the labels. The following formula is used to evaluate the similarity of the two URIs.

$$\text{Sim}(\text{uri}_1, \text{uri}_2, \text{source}) = \frac{\text{Puri}_1, \text{uri}_2}{\text{Puri}_1} + \frac{\text{Puri}_1, \text{uri}_2}{\text{Puri}_2}$$

Ranker also uses the Wikipedia related information in the DBpedia. If there is a hyper textual link in a Wikipedia document w1 to another Wikipedia document w2, there is a corresponding dbpprop:wikilink in DBpedia from corresponding

resource uri1 to uri2. If there is a dbpprop:wikilik between the two uris, there is a stronger relation between the two.

The ranker also checks for the two resources uri1 and uri2, if the rdfs:label of uri1 is contained in the dbpprop:abstract of uri2 and vice versa. At the end a weighted sum is calculated, which is given in Figure 4.

$$w_{google} * \text{sim}(\text{uri}_1, \text{uri}_2, \text{google}) + w_{yahoo} * \text{sim}(\text{uri}_1, \text{uri}_2, \text{yahoo}) + w_{bing} * \text{sim}(\text{uri}_1, \text{uri}_2, \text{bing}) + w_{delicious} * \text{sim}(\text{uri}_1, \text{uri}_2, \text{delicious}) + w_{wikiS} * \text{wikiS}(\text{uri}_1, \text{uri}_2) + w_{abstract} * \text{abstractS}(\text{uri}_1, \text{uri}_2)$$

Fig. 4: Example of a figure caption. (figure caption)

4.3 Context Analyzer

The third main component of this architecture is a context analyzer. Its purpose is to identify a subset of DBpedia nodes which represent a context of interest. The context is represented by the most popular Wikipedia categories, which are reached more often during the graph exploration. These categories are more connected. The context is represented by C. Once we have found the context, we evaluate each newly found resource by comparing it with the most popular DBpedia categories in C. If the score is more than the Threshold=4.0, the resource is considered to be within the context. To compute the context, the DBpedia graph is explored from an initial meaningful set of resources that belong to the selected domain and are selected by the domain experts. The optimal no. of initial resources is 7 [16].

V. Results

Using DBpedia properties, we can examine the semantic similarity between two concepts. In this approach, we have used the skos:subject property of DBpedia. In the following example, we tried to find the similarity between HP Envy, Toshiba 1200 and Toshiba 3100. We want to see if Toshiba 1200 is similar to HP Envy or to Toshiba 3100. Followings are the queries and their corresponding results:

5.1 HP Envy

SPARQL Query: SELECT ?value WHERE { {http://dbpedia.org/resource/HP_Envy dcterms:subject ?value } }

SPARQL results:
Value
:Category :Laptops

Fig. 5: SPARQL Result

5.2 Toshiba 1200

SPARQL Query: SELECT ?value WHERE { {http://dbpedia.org/resource/Toshiba_T1200 dcterms:subject ?value } }

SPARQL results:
Value
:Category :Laptops
:Category :Toshiba_brands

Fig. 6: SPARQL Result

5.3 Toshiba 3100

SPARQL Query: SELECT ?value WHERE { {<http://dbpedia.org/resource/Toshiba_T3100> dcterms:subject ?value } }

SPARQL results:
Value
:Category :Laptops
:Category :Toshiba_T1200

Fig. 7: SPARQL Result

From the above results, we can see that Toshiba 1200 and Toshiba 3100 have more categories in common. They both come under the categories Laptops and Toshiba_brands. But HP Envy comes under Laptops category only but not under Toshiba_brands. So

Toshiba 3100 and Toshiba 1200 are more similar to each other than to HP Envy. We used the above mentioned property for the resources Floppy Disc, Blue-ray Disc and DVD, and their results are shown in figure 8.

Floppy Disc	Blue-ray Disc	DVD
value	value	value
:Category :American Inventions	:Category :High-end Audio	:Category :Computer Storage Media
:Category :Computer Storage Media	:Category :Computer Storage Media	:Category :Joint ventures
:Category :Floppy disk computer storage	:Category :Java Product	:Category :DVD
:Category :Legacy hardware	:Category :Blue ray Disk	:Category :Audio storage
:Category :	:Category :DVD	:Category :120 mm discs
	:Category :Audio Storage	:Category :1995 introductions
	:Category :High definition Television	:Category :Video Storage
	:Category :Video Storage	:Category :Consumer electronics
	:Category :Consumer electronics	:Category :
	:Category :2006 introductions	

Fig. 8: Floppy disc, Blu-ray Disc and DVD

From the above figure, we can see that Blue-ray Disc and DVD are more closely related with each other than with Floppy Disc. Floppy Disc, Blue-ray Disc and DVD

VI. Conclusion

In this paper, we have proposed a technique that will help in semantic schema matching. We also discussed how using the DMOZ directory and the skos:subject property of DBpedia, we can find the similarity between different schemas. In the future, we will try to explore some more DBpedia properties that can help get more

useful data about the DBpedia resources, which will eventually help in semantic schema matching. In this study, we digged into the mappings between the two schemas; whereas this mapping is characterized by their underlying semantics using semantic similarity measure. Moreover, we have articulated taxonomy of all potential semantic similarity measures. Consequently, this study has enabled us to deliver an approach exploiting semantic relations which resides in the DBpedia dataset while utilizing a hybrid ranking system to explain the similarity between nodes of two graphs.

References

- [1] Batini, C., M. Lenzerini, and S.B. Navathe, "A Comparative Analysis of Methodologies for Database Schema Integration," *ACM Computing Surveys*, Vol. 18, No. 4, 1986.
- [2] Vermeer, M. *Semantic Interoperability for Legacy Databases*. Ph.D. thesis, Department of Computer Science, University of Twente, Enschede, Netherlands, 1997.
- [3] H. H. Do and E. Rahm. COMA - a system for flexible combination of schema matching approaches. In *Proceedings of the Very Large Data Bases Conference (VLDB)*, pages 610–621, 2001.
- [4] E. Rahn and P. A. Bernstein. A survey of approaches to automatic schema matching. *Very Large Database J.*, 10(4):334–350, 2001.
- [5] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. *Journal on Data Semantics*, IV:146-171, 2005.
- [6] Buhwan Jeong, Daewon Lee, Hyunbo Cho, Jaewook Lee, A novel method for measuring semantic similarity for XML schema matching, *Expert Systems with Applications*, Vol. 34, Issue 3, 2008, pp. 1651-1658
- [7] S. Melnik, H. Garcia-Molina, E. Rahm, Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching, *Proceedings of the 18th International Conference on Data Engineering*, 2002, pp. 117-128
- [8] Aminul Islam , Diana Inkpen , Iluju Kiringa (2008). Applications of corpus-based semantic similarity and word segmentation to database schema matching, *The VLDB Journal - The International Journal on Very Large Data Bases*, v.17 n.5, p.1293-1320, August 2008
- [9] Allison, L., Dix, T.: A bit-string longest-common-subsequence algorithm. *Information Processing Letters* 23 (1986) 305-310
- [10] Islam, A., Inkpen, D.: Second order co-occurrence pmi for determining the semantic similarity of words. In: *Proceedings of the International Conference on Language Resources and Evaluation*, Genoa, Italy (2006) 1033-1038
- [11] Duchateau F., Bellahsene Z., Roche M., " A Context-based Measure for Discovering Approximate Semantic Matching between Schema Elements", *RCIS*, p. 9-20, 2007.
- [12] F. Duchateau, Z. Bellahsene, M. Roantree, and M. Roche. An Indexing Structure for Automatic Schema Matching. *SMDB-ICDE: International Workshop on Self-Managing Database Systems*, 2007.
- [13] Bing Tian Dai, Nick Koudas, Divesh Srivastava, Anthony K. H. Tung, and Suresh Venkatasubramanian, "Validating Multi-column Schema Matchings by Type," *24th International Conference on Data Engineering (ICDE)*, 2008.
- [14] Jeffrey Partyka, Latifur Khan, Bhavani Thuraisingham, "Semantic Schema Matching Without Shared Instances," *IEEE International Conference on Semantic Computing*, 2009.
- [15] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, "Semantic matching," In *1st European semantic web symposium (ESWS'04)*, pages 61–75, Heraklion, Greece, 2004.
- [16] Roberto Mirizzi, Azzurra Ragone, Tommaso Di Noia, and Eugenio Di Sciascio, Semantic tags generation and retrieval for online advertising.
- [17] L. Page, S. Brin, R. Motwani, and T. Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical report, 1998.
- [18] L. Ding, T. Finin, A. Joshi, R. Pan, S. R. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: a search and metadata engine for the semantic web. In *CIKM '04*, pages 652{659, 2004.
- [19] A. Hogan, A. Harth, and S. Decker. *ReConRank: A Scalable Ranking Method for Semantic Web Data with Context*. 2006.
- [20] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668{677, 1998
- [21] A. Harth, S. Kinsella, and S. Decker. Using naming authority to rank data and ontologies for web search. In *International Semantic Web Conference*, 2009.
- [22] Carmel domshlak, avigdor gal, and haggai roitman, "Rank Aggregation for Automatic Schema Matching," *IEEE transactions on knowledge and data engineering*, vol. 19, no. 4, april 2007.
- [23] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. *Dbpedia - a crystallization point for the web of data*. *Web Semantics: Science, Services and Agents on the World Wide Web*, July 2009.
- [24] P. Chirita, W. Nejdl, R. Paiu, and C. Kohlschuetter. Using ODP metadata to personalize search. In *Proceedings of ACM SIGIR '05*, 2005.

Authors' Profiles

Saira Gilani: Female, Islamabad, Pakistan, received her M.IT degree in Information Technology from Balochistan University, Quetta, Pakistan in 2004. She is an MS student at M.A. Jinnah University, Islamabad, Pakistan and is a member of Center of Research in



Networks and Telecommunication (CoReNeT). Her current research activities include investigation of MAC layer schemes, network security for Vehicular Ad Hoc Networks and data mining and semantic computing.

Muhammad Naeem: Male, Islamabad, Pakistan, Research scholar at department of computer science, M. A. Jinnah University Islamabad Pakistan. His research area is machine learning, semantic computing, text retrieval and data mining.

Raja Habibullah: Male, Islamabad, Pakistan, PhD candidate at department of computer science, M. A. Jinnah University Islamabad Pakistan. His research area are digital libraries, web technologies, semantic computing and ontologies. He can be reached at rajahabib@gmail.com



Amir Qayyum: Male, Islamabad, Pakistan, Dr. Amir Qayyum received his Bachelors in Electrical Engineering from U.E.T., Lahore, Pakistan in 1991, then M.S. in Computer Engineering from E.S.I.M., France in 1995, then

D.E.A. from University of Paris-Sud, France in 1996 and then Doctorate from University of Paris-Sud, France in 2000. He is currently a professor at M. A. Jinnah University, Islamabad and also the head of the Center of Research in Networks and Telecommunication (CoReNeT). His current research interests include mobility management framework for mobile devices, QoS and scalable multicast in IPv6 networks, MANETs and VANETs, VoIP and network security. Currently he is Dean of Faculty of Engineering at Mohammad Ali Jinnah University Islamabad Pakistan.