Research Journal of Applied Sciences, Engineering and Technology 8(4): 460-470, 2014
DOI:10.19026/rjaset.8.994
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: January 20, 2014 Accepted: February 06, 2014 Published: July 25, 2014

Research Article

Utilizing WordNet and Regular Expressions for Instance-based Schema Matching

Ahmed Mounaf Mahdi and Sabrina Tiun
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Selangor, Malaysia

Abstract: Instance-based matching is the process of finding the correspondence of schema elements by comparing the data from different data sources. It is used as an alternative option when the match between schema elements fails. Instance-based matching is applied in many application areas such as website creation and management, schema evolution and migration, data warehousing, database design and data integration. Sometimes the schema information such as (element name, description, data type, etc.) is unavailable or is unable to get the correct match especially when the element name is abbreviation, therefore, if the schema matching failed, the next step is to focus on values stored in the schemas. For these reasons, many recent approaches focus on instance-based matching. In this study, we propose an approach that combines the strength of pattern recognition utilizing regular expressions for numerical domain as well with WordNet for string domain by getting the similarity coefficient in the range of [0,1]. In previous approach, the regular expression is achieved with a good accuracy for numerical instances only and is not implemented on string instances because we need to know the meaning of string to decide if there is a match or not. The using of WordNet-based measures for string instances should guarantee to improve the effectiveness in terms of Precision (P), Recall (R) and F-measure (F). This approach is evaluated with real dataset and the results are found better than using just equality measure for string especially if the schemas are disjoint. The approach achieved 95.3% F-measure (F).

Keywords: Instance-based matching, regular expression, schema matching, WordNet

INTRODUCTION

Database schema is a structure of database that describes the arrangement of its instances, relationships and constraints (Gillani *et al*., 2013).

The application of database schema is important when it is required to integrate different database applications. The problem that will arise when we integrate two different databases is heterogeneity. This heterogeneity divided into two types: structural heterogeneity and semantic heterogeneity. Structural heterogeneity consists of type conflicts, dependency conflicts, key conflicts, or behavioral conflicts; whereas semantic heterogeneity includes semantic conflicts, which is the differences between the databases that are related to the semantic meaning and the planned meaning of data. In order to solve this heterogeneity problem, schema matching is needed (Gillani *et al*., 2013).

Schema matching is a process of identifying the semantic correspondences between elements of the many database schemas (Li and Clifton, 1994; Milo and Zohar, 1998; Madhavan *et al*., 2001; Gillani *et al*., 2013).

In this approach we focus on schema matching problem which is the first step in schema integration task. The schema matching is the process of finding semantic relationships between schema elements that existing in distinct data sources. It can be defined as the process that its input is two schemas and returns a mapping that identifies corresponding elements in the two schemas. Instance-based matching is used to increase the accuracy especially in some cases like when the element names in abbreviation form. Therefore recently, concerns have been put on instance-based matching (Madhavan *et al*., 2001; Gomes de Carvalho *et al*., 2012; Mehdi *et al*., 2012).

Instance-based matching is needed in many applications, such as data and schema integration, e-commerce, evolution of schemas and applications, warehousing, designing of databases, creation and management of websites and component-based development (Rahm and Bernstein, 2001). Suppose two companies decided to corporate with each other; in this case, they need to integrate their databases. As it is known that every company has documents stored in the databases with different schemas and to integrate these schemas, the detecting of the matched candidates is needed for the merging process (Shvaiko and Euzenat, 2005).

Consider a comparison of shopping website is needed. In this case we need to collect the product

Corresponding Author: Ahmed Mounaf Mahdi, Faculty of Information Science and Technologi, Universiti Kebangsaan Malaysia, Selangor, Malaysia

offers from multiple independent online stores and perform the comparison process. The comparison site developers need to match the product categories of each store against the categories of the other stores. For instance, the product_code field in one schema may match the product_ID fields in the other schema. Also the merger between two companies, both of which need to combine their relational databases that propagated by different departments. In integration applications and in other data warehousing applications, matching process of relational schemas is required. Schema matching is used for a variety of other types of schemas such as UML class taxonomies, ER diagrams and ontologies (Melnik *et al*., 2002).

The existing approaches in schema matching classified into three levels:

- Schema level which is using structural schema information
- Instance level which is using a stored data instances
- Hybrid which combines information from schema structure and stored instances (Rahm and Bernstein, 2001)

Sometimes, the schema information (element name, data type, description, etc.) is not available or is not possible to get the correct matching, especially when the element name is abbreviation, therefore, if the schema matching failed, the focus will be on values stored in the schemas. For these reasons, many recent approaches focus on instance-based matching (Mehdi *et al*., 2012; Gomes de Carvalho *et al*., 2012).

Most approaches in instance-based schema matching (Tejada *et al*., 2002; Zaiß *et al*., 2008) used the similarity metrics to measure the similarity between elements and detect the match if exists. Mehdi *et al*. (2012) used the regular expression (regex) to find the correspondences of elements. The process of instance matching using regular expression achieve with a good accuracy for numerical and mixed data instances because the data can be described using a specific pattern, but it is not possible to apply the regex on string domain. The previous approach (Mehdi *et al*., 2012) used the regex for matching numerical instances only, while for the elements with the string data type, a tokenizing process is implemented by considering the first token only for each instance. This will generate a problem of detecting match of non-match strings, such as hot dog will match hot. In addition, it will not match the instances that have the same meaning, such as car will not match automobile and also for cities such as Los Angeles will not match New York (Mehdi *et al*., 2012; Zapilko *et al*., 2012).

LITERATURE REVIEW

Most of previous works (Tejada *et al*., 2001; Tejada *et al*., 2002; Bilenko *et al*., 2003; Duchateau *et al*., 2006; Zaiß *et al*., 2008; Rong *et al*., 2012) have used similarity metrics techniques to find the similarity of instances. These metrics have been classified into two categories:

- Character-based
- Token-based similarity measures

Character-based measure is useful for typographical errors and useless for recognizing the rearrangement of words such as data analyzing and analyzing data. This measure is elaborated as Edit distance, Jaro distance and Q-gram. Edit distance metric is used by Levenshtein (1966), where the measure depends on the number of edit operations insert, delete and replace characters that transformed the string into another string. Jaro distance metrics depends on the number of common characters in the two strings (Jaro, 1989). Finally, Q-gram metric depends on the sequence of N characters for comparing two strings (Moreau *et al*., 2008).

Token-based similarity is useful for recognizing the rearrangement of words and is implemented by breaking the strings into substrings. Jaccard, Atomic strings and Cosine similarity are examples of token-based similarity. Jaccard (1912) proposed a technique to compute the similarity of distributions of Flora in distinct geographical areas. Monge and Elkan (1996) proposed a technique to match two atomic strings. The matching between the two atomic strings will be success if they are equal or one of the two strings is a prefix of the other. The number of matching atomic strings can be divided by the average number of atomic strings to find the similarity between two elements (Elmagarmid *et al*., 2007). Cosine similarity, this technique solved the problem of recognizing the rearrangement of words that mentioned above, so it can consider the words analyzing data and data analyzing are similar. The drawback of this similarity measure is the limitation of solving the spelling error issues. For example, deta analyzing will not be similar to analyzing data (Elmagarmid *et al*., 2007).

Neural network is used for matching task. Yang *et al*. (2008) proposed a Content-Based Schema Matching Algorithm (CBSMA) which is utilizing neural network technique to perform the matching process. CBSMA consists of two steps: Firstly, analyzing the data pattern for calculating the matching pairs by training a group of neural networks. Secondly, rule-based algorithm has been implemented to filter the candidates for getting the correct matching results. This approach has achieved 96% Precision (P) and 90% Recall (R) which are better than other approaches that Yang *et al*. (2008) compared with, which ranged from 65 to 88.9% for Precision (P) and 68.9 to 73% for Recall (R).

Li and Clifton (2000) proposed a Semantic Integration approach (SEMINT) that depends on

constraints and data contents to find the correspondences elements with 1:1 cardinality matching. They used neural networks techniques to learn how the metadata describes the semantics of the elements in a specific domain. The domain knowledge is learned directly from the database. This approach extracts metadata (constraints and instances) from two databases. The metadata forms patterns, that describing the elements, are used as training data for neural networks for recognizing of patterns task. After the neural network trained, the identifying of correspondences elements can be done based on the patterns of elements. Therefore, the system follows two steps: the first is the clustering process implemented on the elements of one schema. The second is the training of a neural network on the cluster centers to provide the most relevant cluster of the second schema. The authors implemented their approach (SEMINT) on real database integration problem and they achieved a Precision (P) with 75% and Recall (R) with 90%.

Doan *et al*. (2001) proposed a system called LSD (Learning Source Description). This approach exploits machine learning techniques to find a match. First, the system asks the user to provide a semantically correspondences of small set of data sources, these correspondences are used with the sources to train a group of learners. Each learner employs a type of information of the source schema or the data itself. After the learning process is finished, LSD system will find the semantic correspondences of new data sources. The authors also extended the machine learning technique to incorporate domain constraint as a source of knowledge and developed a learner that exploits the structural information in XML documents. After this extended, the matching accuracy has increased. For evaluation task, the authors evaluated their approach by several experiments on several real world datasets and they approved that their approach has achieved a high accuracy which is ranged from 71 to 92% for different domains.

Berlin and Motro (2001) discuss system, called Automatch, which is based on Bayesian learning, this approach gains probabilistic knowledge from attribute dictionary which is a knowledge base that created by experts. This dictionary distinguished different elements by their values and the probability guesses of these values. This dictionary also contains attribute names and string patterns. Using probabilistic methods, made the approach match each element of schema A with each element of schema B, with individual scores. After that, for finding the optimal matching between the two schemas, an optimization process relied on a Minimum Cost Maximum Flow network algorithm has been implemented; Automatch exploits the technique of feature selection, to learn a representation of the examples in order to solve the problem of large attribute domains that caused large dictionaries. In this case, the learning will be on a very small subset. This approach show performance that exceeds 70% F-measure but user decision is necessary in order to complete the matching task.

Liang (2008) proposed a domain-independent approach for schema matching. The approach consists of two steps which are: Firstly, computing the mutual information of every two of attributes regardless of the domain information. Mutual information is quantified factor to measure how much sharable information between attributes exists. The author converted a schema into undirected weighted graph. The weight of link between node A and B represents the mutual information between attributes A and B. Secondly, executing of graduated assignment graph matching algorithm to find the correspondence of vertices between graphs. The author evaluated this approach on two datasets and found that the approach is achieved 70% precision on average.

Gomes de Carvalho *et al*. (2012) proposed an approach to solve the problem of automatically finding one to one as well as many to many matching between schemas using only the data instances. The authors depend on matching techniques that are based on genetic programming. For the fitness function which is evaluated in each step, they proposed two strategies which are entity-oriented and value-oriented strategies to find the matching elements. They evaluate their approach using real and unreal datasets. The results show that the approach can find one to one as well as complex matches. The authors achieved 57% accuracy to find 1-1 matches on partially overlapped data and 100% accuracy to find 1-1 matches on fully overlapped data. Also they worked on three disjoint datasets and they achieved from 42 to 85% accuracy.

Mehdi *et al*. (2012) proposed an approach to find the correspondences of instances by using regular expression. Their approach generates the regex list automatically and finds the correspondence of numeric and mixed columns by matching the regex which is generated for a specific column in first schema, with the columns of second schema. For the string values the authors take a first token of the attribute to compare with the first token of strings existing in the other schema. They depend on equality measure of two strings regardless the meaning of these strings. The approach has been evaluated with series of experiments and the results achieved 98% accuracy which is better than other string similarity metrics that the authors compared with, such as; LCS, which its accuracy is 95.90%.

Zapilko *et al*. (2012) also utilized the regular expression to solve the matching problem. The authors define a list of regex which describe different elements such as a purely numeric element or mixed data element. The approach is achieved a good result for matching numeric and mixed instances.

Yatskevich and Giunchiglia (2004) proposed an approach that utilize WordNet as a knowledge source for getting the semantic relations of two concepts instead of similarity coefficient with values [0, 1]. The authors present twelve element level matchers which

utilize WordNet to get the semantic relation. They evaluated their approach with other matching systems and the results were comparable with 42% Precision (P) and 58% Recall (R).

METHODOLOGY

The framework: Figure 1 shows the framework of the proposed approach and how the steps of this approach are organized.
The framework is organized as the following steps:

- Prepare the dataset by converting the text files which includes the instances of schemas into two dimension array
- Identifying the data type of each column to find out whether a specific column is a string, numeric or mixed data
- Select the samples randomly from each column
- Perform the matching on the samples selected, with regex if the data type of the samples is numeric or mixed and with WordNet if the data type is string
- The final output will be the matched elements

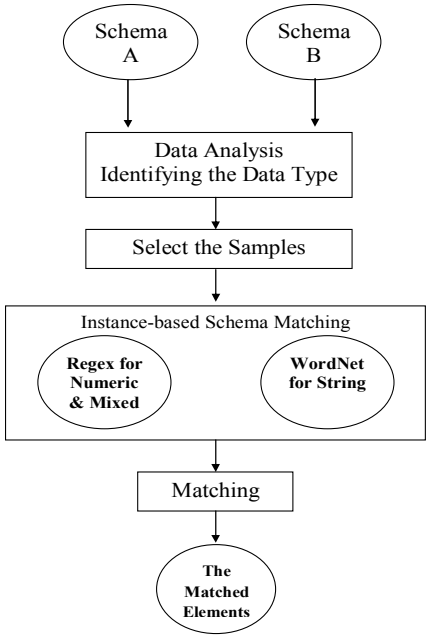

Fig. 1: Framework of proposed approach

Our approach used two techniques for matching string elements as well numeric and mixed elements. We used WordNet to find the similarity of string instances and utilized regular expressions for numeric and mixed instances to find the elements that match the same regular expression. We discusses about these two methods in the next subsections.

WordNet: There are many techniques used to find the similarity or relatedness between two concepts. In this study we tried to find the best technique to enhance the matching of two string type elements. From the literature review, we have found that the character-based similarity measures and token-based similarity measures are not suitable for matching if there are no shared characters between the two comparing concepts.

The problem of finding the correctly matched elements is not a trivial to be solved because of structure variety and semantic diversity of data. Some auxiliary sources such as dictionary and thesauri can help to reduce the degree of difficulty (Liang, 2008).

In recent years, several concerns have been put on measuring based on WordNet (Yatskevich and Giunchiglia, 2004; Varelas *et al*., 2005; Lin and Sandkuhl, 2008; Meng *et al*., 2013). For this reason we utilized the WordNet in this study to help us to find the similarity between two concepts.

WordNet is the product of research project that performed at Princeton University (Miller and Fellbaum, 1998). WordNet includes three databases; the first is for nouns, the second is for verbs and the third for adjectives and adverbs. WordNet also includes a set of synonyms which are also called synsets. A synset represents a concept or a sense of a set of terms. Synsets produce different semantic relationships such as synonymy which is the similar relationship and antonymy which is the opposite relationship, hypernymy/hyponymy which are super concept/sub concept relationship also called Is-A hierarchy/taxonomy, meronymy which is part-of relationship and holonymy which is has-a relationship. The semantic relations through the synsets are varies depending on the grammatical category. WordNet also produces some descriptions of each concept (gloss) including definitions and examples.

Semantic similarity measures are used for implementing some tasks such as term disambiguation (Patwardhan *et al*., 2003), text segmentation (Kozima, 1994) and for consistency of ontologies. Many measures have been proposed, all measures are categorized by Meng *et al*. (2013) into four categories: path length-based measures, information content-based measures, feature-based measures and hybrid measures. In the next subsections we will introduce a background of all measures that will be used in our experiments.

Path-based measures: The main notion of path-based measures is that the similarity of two strings is a function of the length of the path that links the first string with the second and the position of these strings in the taxonomy.

In this study, we implemented two methods of path-based measures which are; shortest path-based measure and Wu and Palmer (1994) measure, because of, they have lower and upper similarity values and ranged between 0 and 1.

Shortest path-based measure: The measure only takes the length of the shortest path from c_1 to c_2 *len* (c_1, c_2) into considerate. It is a variant of two distance method

(Rada *et al*., 1989; Bulskov *et al*., 2002). In this measure, the similarity of two concepts *Sim* (c_1, c_2) depends on the length of path that connects the two concepts in the taxonomy and how much the concepts are close to each other. The similarity value in this type of measure is ranged between 0 and 1:

$$Sim_{path}(c_1, c_2) = 2 * deep\_max - len(c_1, c_2) \quad (1)$$

deep_max value depends on the version of WordNet and it is a fixed value. The similarity of two concepts (c_1, c_2) is the function of the length of the shortest path from c_1 to c_2 which is represented by *len* (c_1, c_2). Thus, if *len* (c_1, c_2) = *len* (c_3, c_4) this leads to $sim_{path}(c_1, c_2) = sim_{path}(c_3, c_4)$.

Wu and Palmer's measure: Wu and Palmer (1994) presented a measure that focuses on the position of concepts c_1 and c_2 in the taxonomy relatively to the position of the most specific common concept *lso* (c_1, c_2) into account. The similarity value also is ranged between 0 and 1:

$$Sim_{wp}(c_1, c_2) = \frac{2*depth(lso(c_1,c_2))}{len(c_1,c_2)+2*depth(lso((c_1,c_2)))} \quad (2)$$

The similarity of two concepts (c_1, c_2) is the function of the distance of each concept and the lowest common *subsumer* (*lso* (c_1, c_2)). therefore, if *len* (c_1, c_2) = *len* (c_3, c_4) and *lso* (c_1, c_2) = *lso* (c_3, c_4) this leads to $Sim_{wp}(c_1, c_2) = Sim_{wp}(c_3, c_4)$.

Information content-based measures: This measure considers that every concept has a lot of information in WordNet. Similarity measures are relying on the information content of the concept. If there is much common information between the concepts, then the two concepts have the same meaning.

In this study, we implemented Lin's measure that its similarity ranged between 0 and 1.

Lin's measure: Lin (1998) proposed a similarity measure that uses both the information content that subsumes the concepts in taxonomy and the information needed to fully describe these concepts. The similarity values of this measure are ranged between 0 and 1 as same as a shortest path-measure and Wu and Palmer's measure:

$$Sim_{Lin}(c_1, c_2) = \frac{2*IC(lso(c_1,c_2))}{IC(c_1)+IC(c_2)} \quad (3)$$

There is no standard to evaluate the effectiveness of semantic similarity measures. In this study we depend on application-oriented evaluation (Blanchard *et al*., 2006; Budanitsky and Hirst, 2006). This means, if a specific application requires a measure of semantic similarity, we have to implement many measures and compare the performance of each measure separately to find the most effective measure for a specific area (Meng *et al*., 2013). We have chosen three measures; the selection of these measures is depending on two observations:

- Path based-measures and information content-based measures perform better than feature-based and hybrid measures (Petrakis *et al*., 2006). Therefore, we have chosen the three measures from these two categories.
- Some of measures do not have upper bound of similarity. Therefore, for this paper we have chosen the measures that have lower and upper bound and ranged between 0 and 1.

WORDNET::SIMILARITY

WordNet::Similarity[1] is a software package developed at the University of Minnesota as an open source software for Perl. It helps the user to find the semantic similarity or the relatedness between two concepts. This system provides six similarity measures and three relatedness measures based on the WordNet database (Fellbaum, 1998). The similarity measures are based on is-a hierarchy. These measures are divided into only two group path-based measures and information content based measures, however it does not include feature-based measure. For our approach, we used WordNet Similarity for Java[2] (WS4J), which provides a Java API of Princeton's English WordNet. It is a re-implementation of Wordnet::similarity for Perl that mentioned above.

Using WordNet for matching: We have built a function that calculates the similarity of two items (S_1, S_2). The items are the current item from the source schema and every string item of the target schema. The items are sent as a one token (S_1, S_2) to compare a compound words that has a specific meaning as a one concept such as some cities like Los Angeles. Also we sent the items as a list of tokens (tokens (S_1), tokens (S_2)). For example, if S_1 is Los Angeles and S_2 is New York, the system will calculate the similarity of Los Angeles with New York will find a high similarity, Los with New will not find a similarity, Los with York will not find a similarity, Los with New will not find a similarity, Angeles with New will not find a similarity, Angeles with York will not find a similarity.

Another example if S_1 is American and S_2 is American new, the calculation will be: American with American new will not find a match, American with American will find a match, American with New will not find a match. The algorithm *CalcWordNet* (Algorithm 1) illustrates the function that calculates the similarity of two items.

The algorithm *CalcWordNet* (Algorithm 1) shows the procedures of finding the similarity of two items. We can choose one of the WordNet based measures such as; Wu and Palmer's measure as it is illustrated in line 10 or a combination of measures by adding some codes as it is shown in line 13 in Algorithm 1. After that, we defined a specific threshold that similarity has to exceed it to consider that the elements are matched. Let's say the threshold is 0.8. In this case, if the similarity of two items is more than or equal to 0.8, the possibility of considering the matched elements will be increased by one. The possibility of considering the first column of the source schema match the first column of the target schema is a value of degree [0] [0]. Therefore, the possibility of considering the column (i) in source schema match the column (j) in target schema is represented by the value of degree [i] [j]. In this array, (i) represents the index of column of the source schema and the (j) represents the index of column of the target schema.

After we finish from checking processes of all samples which are 43 samples (10% of the number of rows in restaurant dataset) in our approach. We call a function that finds the highest value in every row of degree array. The column of the highest value represents the column of target schema which matched with a column in a source schema. Finally, the matched elements are printed.

Algorithm 1: *CalcWordNet*

1. Pass in: S1: represents a token from Schema 1
2. S2: represents a token from Schema 2
3. pos: represent the part of speech for these tokens. We use noun only
4. i: represents index of the current column of schema 1
5. j: represents index of the current column of schema 2
6. Pass out: degree array
7. Let sense ←1 which is the most common sense.
8. Let degree be a two dimensional array for the similarity degree of two elements
9. Let degree [][]←0
10. If *wup* (*S*1, *sense*, *S*2, *sense*, *pos*) ≥0.8 then
11. degree [i][j]←degree [i][j] +1
12. End if
13. If *lin* (*S*1, *sense*, *S*2, *sense*, *pos*) ≥0.76 then
14. degree [i][j]←degree [i][j] +1
15. End if

Regular expression: For numeric and mixed data such as: age, address, date and so on it is good to use a pattern recognition technique to describe the format of text. Regular expression is a good choice for this purpose and it is used in instance-based matching with good result for numeric and mixed data (Mehdi *et al*., 2012; Zapilko *et al*., 2012).

Regular expressions (regexps or regex) are the key to powerful, flexible and efficient text processing. It is used for matching a specific string of text and it can be defined as a string consists of a number of characters and meta characters (*, +, $, ^, and ?), for example, [0-9]+ matches any group of numbers, [a-z]+ matches any collection of lowercase letters (Spishak *et al*., 2012; Mehdi *et al*., 2012). For more details about the regex (Friedl, 2006).

The need to search for regular expressions arises in many text-based applications such as text retrieval, text editing, computational biology and network security. In computer security virus or spam signatures (Kumar *et al*., 2007; Xie *et al*., 2008) are usually represented by regular expressions, since they permit to handle many different variations of the same virus or to handle similar spam patterns (Belazzougui and Raffinot, 2012).

Regular expressions have many strength points that make us choose it in this approach, which are (Doan and Halevy, 2005):

- Regular expression captures valuable user knowledge about the domain quickly and concisely.
- Regular expressions do not need any training or learning as learning techniques. Therefore, it is inexpensive method.

Regular expressions are used by many text editors and utilized to search and manipulate bodies of text based on certain patterns. Many programming languages support regular expressions for string manipulation. For example, Java,.NET languages and C++ provides a support to deal with regex in its standard library. In this study we used Java to achieve our purposes.

Using regular expressions for matching: For numeric and mixed data, a function has been built to compare the two items (one item from the source and the other from the target schema) with a list of regular expressions, if the two items match the same regex, this means the elements that these values come from are matched. The regular expressions list has been predefined regarding to the data which we have in Restaurant dataset as the following:

- For mixed data, the regex is [0-9 A-Za-z.] +or '[0-9]+'
- For phone NO, the regex is ([0-9]{3}[-/]){2}[0-9]{4}

The same work of degree [i] [j] that is used in matching by WordNet has been used here with name *degreeReg* and also we find the maximum value of each row to select the correct correspondences from the list of candidates. The algorithm below *MatchRegex* (Algorithm 2) shows the steps of finding the correspondence elements by checking the items with the regex list.

Algorithm 2: *MatchRegex*

```
1.  Pass in:   S1: represents a token from Schema 1
2.             S2: represents a token from Schema 2
3.             i: represents index of the current column
                  of schema 1
4.             j: represents index of the current column
                  of schema 2
5.  Pass out: degreeReg
6.   Begin
7.       Let RegList be an array of regular expressions
8.       For each RegList_i Do
9.         If S1 match RegList_i and S2 match RegList_i
10.              degreeReg_i,j ← degreeReg_i,j +1
11.        End if
12.      End for
13.  End
```

Algorithm 3 is the main algorithm that includes all steps of our approach; the determining of data type of each column which is illustrated in lines 9 to 12, the selection of samples is shown in line 15, the procedures of getting the tokens of instances and calculating the similarity by WordNet if the data type is string is shown in lines 16 to 23 and the instructions sending the instances to *MatchRegex* algorithm to find the elements that have the same regex is shows in lines 25 to 31.

Algorithm 3: Main Algorithm

```
1.  Pass in:   Source schema SS = {A1, A2, …, An }
2.             Target schema TS = {B1, B2, ..., Bm}
3.  Pass out: The matched elements
4.  Begin
5.  Let SSize = The size of samples
6.  Let Nrows = length of Ai
7.  Let LTS = List of tokens of each random value from
    SS
8.  Let TT = List of tokens of each random value from
    TS
9.  For each Ai of SS DO
10.   Type of  Column (Ai)
11. For each Bi of TS DO
12.   Type of Column (Bi)
13. For  i = 0 until SSize DO
14.   For each Aj of SS DO
15.    random = Get a random value between 0 and
       Nrows
16.    IF type of colum_j = "string" THEN
17.      LTS = Get tokens of A_random,j
18.      For each Bk of TS DO
19.        IF type of colum_k = "string" THEN
20.          LTT = Get tokens of B_random,k
21.          CalcWordNet (LTS, LTT)
22.      End if
23.    End for
24. Else
25.      LTS = A_random,j
26.      For each Bk of TS DO
27.        If type of colum_k is not "string" THEN
28.          LTT = B_random, k
29.        MatchRegex (A_random,j, B_random,k)
30.      End if
31.    End for
32.   End if
33.  End for
34. End for
35. End
```

RESULTS AND DISCUSSION

Dataset: To evaluate the performance of any approach a set of experiments on real world datasets should be conducted. Many real world datasets have been used in instance-based schema matching area. Liang (2008) used two datasets: education assessment and book statistics. Whilst Mehdi *et al*. (2012), used only one dataset which is restaurant dataset and both of them horizontally split the datasets into two parts to cover the needs of two schemas from different sources. Restaurant dataset is used also by Tejada *et al*. (2002).

We conduct several experiments on two datasets; restaurant datasets[3] which is collected its information from two websites: Zagat website and Fooder website. Restaurant dataset has 864 records and 6 elements: Name, Address, City, Phone number, Type of food and Class. Name, City and Type of food are string type; Address, Phone number and Class are mixed. We split the dataset horizontally into two parts and consider them as two schemas from distinct sources.

In this experiment we aim to find 1:1 matching between two schemas.

Implementing shortest path-based measure with regex: In the first analysis, we implement the shortest path-based measure for string data and regular expression for mixed data using restaurant dataset. We have chosen the threshold of similarity to be 0.6. This is the best threshold we got it after a series of experiments to find the match. Therefore, if the similarity of two instances is more than or equal to 0.6 the possibility of matching two elements will be increased by one. Figure 2 shows the effectiveness based on Precision (P), Recall (R) and F-measure (F) with 10% of the samples chosen for comparing process.

From Fig. 2 we notice that the shortest path-based measure give as the same as equality measure with 100% Precision (P), 93.2% Recall (R) and 96.2% F-measure (F).

Implementing Wu and Palmer's measure with regex: In this analysis we implemented Wu and Palmer's measure which depends on the position of concepts in the taxonomy relatively to the most specific common concept. We also did a several experiments to

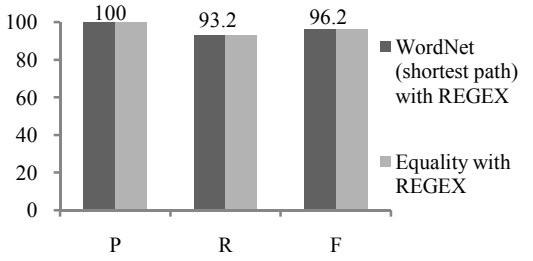

Fig. 2: Comparison using shortest path based measure with regex and equality with regex

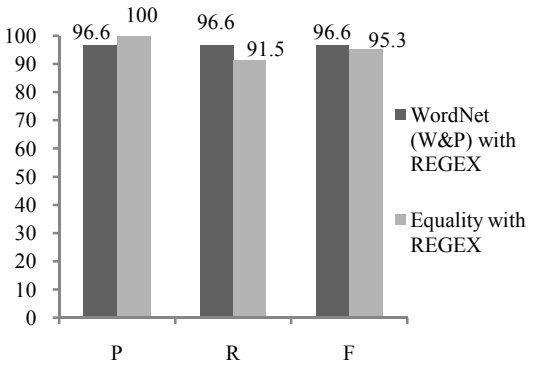

Fig. 3: Comparison using W&P measure with regex and equality with regex

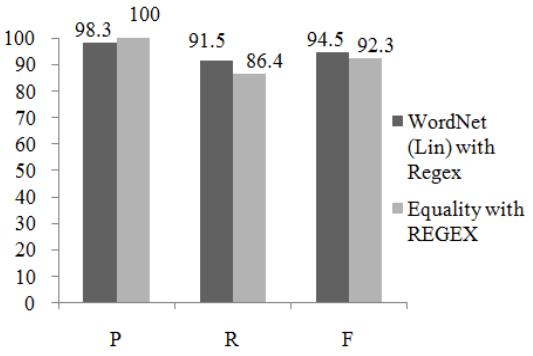

Fig. 4: Comparison using Lin's measure with regex and equality with regex

reach to the best threshold. The best result we got it with threshold equal to 0.8. Figure 3 shows the result in terms of Precision (P), Recall (R) and F-measures (F). We notice that Wu and Palmer's measure has enhanced the Recall by 96.6% comparing with equality measure that the previous work of Mehdi *et al*. (2012) used for string type elements with just 91.5% Recall (R). Therefore, implementing of Wu and Palmer's measure with regex has enhanced the F-measure (F) by 1.3%.

Implementing Lin's measure with regex: In this analysis, we implemented Lin's measure which uses the

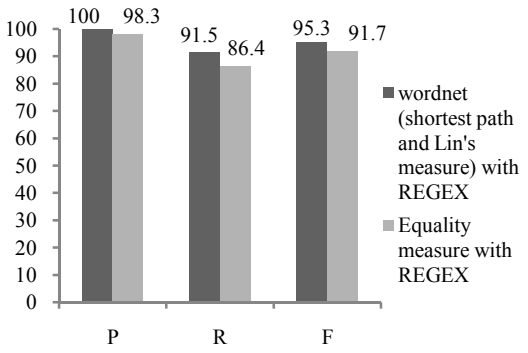

Fig. 5: Comparison of using (shortest path based and Lin's measure with regex) and (equality with regex)

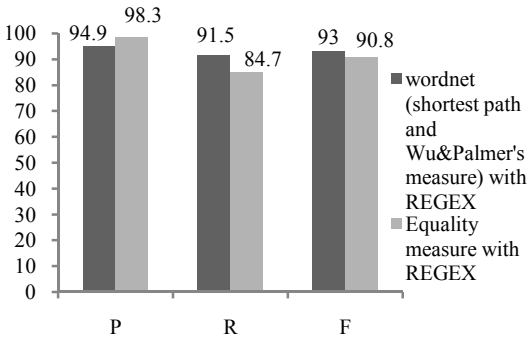

Fig. 6: Comparison of using (shortest path based, W&P measure with regex) and (equality with regex)

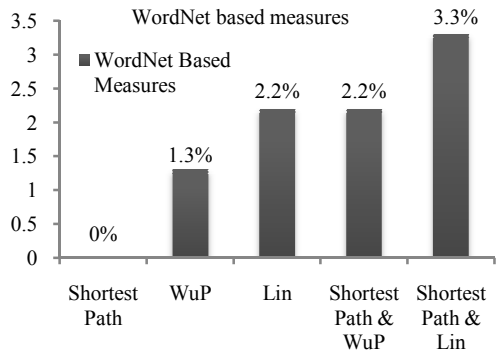

Fig. 7: The difference between WordNet measures and equality measure in %

amount of information which is necessary to state the commonality between the two concepts and the information needed for describing the concepts. In our experiments we have found that 0.76 is the best threshold to give us good result. Therefore if the similarity is equal or greater than 0.76, the possibility of considering the match between two elements will be increased by one. The Fig. 4 shows the result of implementing Lin's measure with regex.

From Fig. 4, it is noted that this measure improved the Recall (R) with 91.5% comparing with 86.4% when

we use equality measure. So the increasing is 5.1% and the F-measure (F) is 94.5% comparing with 92.3% for equality measure.

Combination of shortest path-based measure and Lin's measure with regex: In this analysis we have combined one of the path based measure which is shortest path measure with one of the content based measure which is Lin's measure to evaluate the usefulness of using both of them for string data and regex for numeric and mixed data. We used the same threshold that we have got it when we applied shortest-path based measure which is 0.6 and also the same threshold for Lin's measure which is 0.76. The result is shown in Fig. 5. The combination of these two measures has increased the Precision (P), Recall (R) and F-measure (F) from 98.3, 86.4 and 91.7% respectively when we use the equality measure into 100, 91.5 and 95.3% respectively if we use these two methods (Shortest path-based measure and Lin's measure) (Fig. 5).

Combination of shortest path-based measure and Wu and P measure with regex: In this analysis, we have combined the shortest path based measure with Wu and Palmer's measure using restaurant dataset and use the same threshold that used for the two methods separately. The Fig. 6 shows the result.

Finally, regarding to the results that we obtained, we will show the difference of F-measures (F) between equality measure and our approach that using the regular expression with different WordNet based measures using restaurant dataset in Fig. 7. We noticed that the combination of WordNet measures is useful.

CONCLUSION

In this study we presented an instance-based approach to schema matching. Our approach utilizes two techniques for matching string elements as well numeric and mixed elements with high effectiveness.

We chose regular expression to describe the instances that has a specific form such as numeric instances and mixed data such as the date elements. This technique gives a good result in our experiments. For string data we need to know the semantic meaning to determine the matched elements. We chose WordNet-based measures for this issue.

Our approach achieved the best result when we combined the regular expression with two WordNet-based measures which are: shortest path-based and Lin's measures. The approach achieved Precision (P) with 100%, Recall (R) with 91.5, % F-measure (F) with 95.3%.

For a future work, we will conduct more experiments on other datasets and implement other WordNet-based measures. We will also use the techniques that proposed in this study to solve a problem of finding the matched elements of 1:n, n:1 and n:m schemas, complex matching which is not sufficiently expressive and put much of the matching burden on the user.

REFERENCES

Belazzougui, D. and M. Raffinot, 2012. Approximate regular expression matching with multi-strings. J. Discret. Algorithm., 18: 14-21.

Berlin, J. and A. Motro, 2001. Autoplex: Automated discovery of content for virtual databases. Lect. Notes Comput. Sc., 2172: 108-122.

Bilenko, M., R. Mooney, W. Cohen, P. Ravikumar and S. Fienberg, 2003. Adaptive name matching in information integration. IEEE Intell. Syst., 18(5): 16-23.

Blanchard, E., P. Kuntz, M. Harzallah and H. Briand, 2006. A tree-based similarity for evaluating concept proximities in an ontology. St. Class. Dat. Anal., pp: 3-11.

Budanitsky, A. and G. Hirst, 2006. Evaluating wordnet-based measures of lexical semantic relatedness. Comput. Linguist., 32(1): 13-47.

Bulskov, H., R. Knappe and T. Andreasen, 2002. On measuring similarity for conceptual querying. Lect. Notes Comput. Sc., 2522: 100-111.

Doan, A. and A.Y. Halevy, 2005. Semantic integration research in the database community: A brief survey. AI Mag., 26(1): 83.

Doan, A., P. Domingos and A.Y. Halevy, 2001. Reconciling schemas of disparate data sources: A machine-learning approach. ACM Sigmod Record, 30(2): 509-520.

Duchateau, F., Z. Bellahsene and M. Roche, 2006. A Context-based Measure for Discovering Approximate Semantic Matching between Schema Elements. Retrieved from: hal-lirmm.ccsd.cnrs.fr/docs/00/11/38/49/ PDF/RR-06053.pdf.

Elmagarmid, A.K., P.G. Ipeirotis and V.S. Verykios, 2007. Duplicate record detection: A survey. IEEE T. Knowl. Data En., 19(1): 1-16.

Fellbaum, C., 1998. A semantic network of english: The mother of all WordNets. Comput. Humanities, 32(2-3): 209-220.

Friedl, J., 2006. Mastering Regular Expressions. O'Reilly Media, Incorporated.

Gillani, S., M. Naeem, R. Habibullah and A. Qayyum, 2013. Semantic schema matching using DBpedia. Int. J. Intell. Syst. Appl., 5(4): 72.

Gomes de Carvalho, M., A.H. Laender, M. André Gonçalves and A.S. Da Silva, 2012. An evolutionary approach to complex schema matching. Inform. Syst., 38(3): 302-316.

Jaccard, P., 1912. The distribution of the flora in the alpine zone. 1. New Phytol., 11(2): 37-50.

Jaro, M.A., 1989. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. J. Am. Stat. Assoc., 84(406): 414-420.

Kozima, H., 1994. Computing lexical cohesion as a tool for text analysis. Ph.D. Thesis, University of Electro-Communications.

Kumar, S., B. Chandrasekaran, J. Turner and G. Varghese, 2007. Curing regular expressions matching algorithms from insomnia, amnesia and acalculia. Proceedings of the 3rd ACM/IEEE Symposium on Architecture for Networking and Communications Systems, pp: 155-164.

Levenshtein, V.I., 1966. Binary codes capable of correcting deletions, insertions and reversals. Sov. Phys. Doklady, 10(8): 707.

Li, W.S. and C. Clifton, 1994. Semantic integration in heterogeneous databases using neural networks. Proceedings of the 20th VLDB Conference. Santiago, Chile, pp: 12-15.

Li, W.S. and C. Clifton, 2000. SEMINT: A tool for identifying attribute correspondences in heterogeneous databases using neural networks. Data Knowl. Eng., 33(1): 49-84.

Liang, Y., 2008. An instance-based approach for domain-independent schema matching. Proceedings of the 46th Annual Southeast Regional Conference. Auburn, Alabama, pp: 268-271.

Lin, D., 1998. An information-theoretic definition of similarity. Proceedings of the 15th International Conference on Machine Learning, pp: 296-304.

Lin, F. and K. Sandkuhl, 2008. A survey of exploiting wordnet in ontology matching. Int. Fed. Info. Proc., 276: 341-350.

Madhavan, J., P.A. Bernstein and E. Rahm, 2001. Generic schema matching with cupid. Proceedings of the International Conference on Very Large Data Bases, pp: 49-58.

Mehdi, O.A., H. Ibrahim and L.S. Affendey, 2012. Instance based matching using regular expression. Proc. Comput. Sci., 10: 688-695.

Melnik, S., H. Garcia-Molina and E. Rahm, 2002. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. Proceedings of the 18th International Conference on Data Engineering, pp: 117-128.

Meng, L., R. Huang and J. Gu, 2013. A Review of Semantic Similarity Measures in WordNet. Int. J. Hybrid Inform. Technol., 6(1).

Miller, G. and C. Fellbaum, 1998. Wordnet: An Electronic Lexical Database. MIT Press, Cambridge.

Milo, T. and S. Zohar, 1998. Using schema matching to simplify heterogeneous data translation. Proceeding of the 24th VLDB Conference. New York, USA, pp: 24-27.

Monge, A.E. and C. Elkan, 1996. The field matching problem: Algorithms and applications. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp: 267-270.

Moreau, E., F. Yvon and O. Cappé, 2008. Robust similarity measures for named entities matching. Proceedings of the 22nd International Conference on Computational Linguistics, 1: 593-600.

Patwardhan, S., S. Banerjee and T. Pedersen, 2003. Using measures of semantic relatedness for word sense disambiguation. Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing'03), pp: 241-257.

Petrakis, E.G., G. Varelas, A. Hliaoutakis and P. Raftopoulou, 2006. Design and evaluation of semantic similarity measures for concepts stemming from the same or different ontologies. Proceedings of the 4th Workshop on Multimedia Semantics (WMS'06), pp: 44-52.

Rada, R., H. Mili, E. Bicknell and M. Blettner, 1989. Development and application of a metric on semantic nets. IEEE T. Syst. Man Cyb., 19(1): 17-30.

Rahm, E. and P.A. Bernstein, 2001. A survey of approaches to automatic schema matching. VLDB J., 10(4): 334-350.

Rong, S., X. Niu, E.W. Xiang, H. Wang, Q. Yang and Y. Yu, 2012. A machine learning approach for instance matching based on similarity metrics. Proceedings of the 11th International Conference on the Semantic Web-Volume Part I (ISWC'12).

Shvaiko, P. and J. Euzenat, 2005. A survey of schema-based matching approaches. Lect. Notes Comput. Sc., 3730: 146-171.

Spishak, E., W. Dietl and M.D. Ernst, 2012. A type system for regular expressions. Proceedings of the 14th Workshop on Formal Techniques for Java-Like Programs, pp: 20-26.

Tejada, S., C.A. Knoblock and S. Minton, 2001. Learning object identification rules for information integration. Inform. Syst., 26(8): 607-633.

Tejada, S., C.A. Knoblock and S. Minton, 2002. Learning domain-independent string transformation weights for high accuracy object identification. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp: 350-359.

Varelas, G., E. Voutsakis, P. Raftopoulou, E.G. Petrakis and E.E. Milios, 2005. Semantic similarity methods in wordNet and their application to information retrieval on the web. Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management, pp: 10-16.

Wu, Z. and M. Palmer, 1994. Verbs semantics and lexical selection. Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, pp: 133-138.

Xie, Y., F. Yu, K. Achan, R. Panigrahy, G. Hulten and I. Osipkov, 2008. Spamming botnets: Signatures and characteristics. Comput. Commun. Rev., 38(4): 171-182.

Yang, Y., M. Chen and B. Gao, 2008. An effective content-based schema matching algorithm. Proceedings of the International Seminar on Future Information Technology and Management Engineering (FITME '08), pp: 7-11.

Yatskevich, M. and F. Giunchiglia, 2004. Element level semantic matching using WordNet. Proceeding of the Meaning Coordination and Negotiation Workshop. ISWC.

Zaiß, K., T. Schlüter and S. Conrad, 2008. Instance-based ontology matching using regular expressions. Proceeding of the OTM 2008 Workshops on the Move to Meaningful Internet Systems, pp: 40-41.

Zapilko, B., M. Zloch and J. Schaible, 2012. Utilizing regular expressions for instance-based schema matching. Procedia Comput. Sci., 10: 688-695.

End note:

1 http://www.d.umn.edu/~tpederse/similarity.html
2 https://code.google.com/p/ws4j/.
3 http://www.infochimps.com/datasets/restaurant