Indonesian Journal of Electrical Engineering and Computer Science
Vol. 10, No. 3, June 2018, pp. 1266~1277
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v10.i3.pp1266-1277

An Empirical Comparative Study of Instance-based Schema Matching

Mogahed Alzeber[1], Ali A. Alwan[2], Azlin Nordin[3], Abedallah Zaid Abualkishik[4]
[1,2,3]Department of Computer Science, Kulliyyah of Information and Communication Technolgoy, International Islamic University Malaysia, Malaysia
[4]College of Computer Information Technology, American University in the Emirates, Dubai, United Arab Emirates

Article Info

Article history:

Received Dec 14, 2017
Revised Feb 25, 2018
Accepted Mar 11, 2018

Keywords:

Database instances
Data integration
Google similarity
Regular Expression
Schema matching

ABSTRACT

The main issue concern of schema matching is how to support the merging decision by providing matching between attributes of different schemas. There have been many works in the literature toward utilizing database instances to detect the correspondence between attributes. Most of these previous works aim at improving the match accuracy. We observed that no technique managed to provide an accurate matching for different types of data. In other words, some of the techniques treat numeric values as strings. Similarly, other techniques process textual instance, as numeric, and this negatively influences the process of discovering the match and compromising the matching result. Thus, a practical comparative study between syntactic and semantic techniques is needed. The study emphasizes on analyzing these techniques to determine the strengths and weaknesses of each technique. This paper aims at comparing two different instance-based matching techniques, namely: (i) regular expression and (ii) Google similarity to identify the match between attributes. Several analyses have been conducted on real and synthetic data sets to evaluate the performance of these techniques with respect to Precision (P), Recall (R) and F-Measure.

Corresponding Author:

Ali A. Alwan,
Department of Computer Science,
Kulliyyah of Information and Communication Technology,
International Islamic University Malaysia, IIUM,
P. O. Box 10, 50728 Kuala Lumpur, Malaysia.
Email: aliamer@iium.edu.my

1. INTRODUCTION

Several attempts have been conducted to combine data from different heterogeneous sources to form a unified global view. This process called data integration, which aims to represent data in one single view and facilitates the process of interacting with the data to be appearing as one single information system [1] [2]. However, it is very challenging to integrate and manage data which comes from several different sources that are being independently developed. This is due to the fact that there are different representations of these sources, and data sources might not be designed in a way to adopt the same abstraction principles or have similar semantic concepts to be fully used [3].

There are many reasons for integrating independent heterogeneous information systems into one global information system. For example, many firms might attempt to integrate some of the developed heterogeneous data sources where these businesses have various databases, and each database might consist of a vast number of tables that encompass different attributes. The process of data integration can be performed either manually or semi-automatically. In both approaches, there are some issues that the machine might face during the integration process, including detecting the correspondence between database schemas at the schema level, instance level, or both [2], [4] and [8]. Besides, identifying the conflicts of syntax and semantic

heterogeneity between schemas is also a significant issue during data integration. For this reason, schema matching has been proposed to handle the process of discovering the correspondence between schemas and resolve conflicts when occurred.

However, schema matching approach can only be utilized with standardized database environments where schema attributes names are unambiguous. Therefore, using schema matching is inappropriate when databases are developed separately and without unified standards [9]. Furthermore, it is impractical to employ the schema design information "schema attributes" to determine the correspondences attributes when different abbreviations of attribute names "column's names" is used to represent the same real world entities or objects [3] [9]. There are many real life applications where schema information is unavailable or available but worthless to be used, examples including homeland security, crime investigation, counterterrorism [3, 8, 10]. Thus, in these cases, utilizing the instances is the best available alternative to achieve the schema matching between databases gives a precise characterization of the real contents of schema attributes [11]. Instance-based schema matching attempts to extract the semantic relationship between targeted attributes via their values "instance".

Two different classes for matching have been proposed, namely: syntactic and semantic. The syntactic emphasizes on the heterogeneity in the structure of the table (attributes) to determine the match. While the semantic class focuses on the heterogeneity in the meaning of the instances. Many techniques have been proposed that rely on syntactic, including N-gram, and regular expression. While the most effective techniques that rely on semantic including, Latent Semantic analysis (LSA), WordNet/Thesaurus, and Google similarity. By examining the previous works, we noticed that most of techniques could not achieved precise matching for different data types. In other words, some of the techniques treat numeric values as strings. This negatively influences on discovering the match and deteriorates the quality of match results. Similarly, other techniques treat textual instance, as numeric, and also impact the quality of the match results.

In this paper, we examine two strategies utilizing Google Similarity and Regular expression techniques to identify the semantic match between database attributes using the available instances. The study should carry out extensive experiments that help researchers in this area of research to understand the capabilities and the limitations of each technique.

The rest of the paper is organized as follows. The previous related works are reviewed and reported in section 2. The detail description of the proposed approach for instance-based schema matching has been explained in section 3. The following section 4 reports the results of the experiment. The experiment results have been reported in section 5. The conclusion is presented in section 6.

2. RELATED WORK

Instance-based schema matching has been investigated by numerous studies that concentrate on enhancing the accuracy of the schema matching result [3, 6-7, 12-18]. Different approaches have been proposed, adopted various strategies for precise determination of correspondence between attributes of schemas. Most of the previous works related to schema matching utilized different similarity metrics techniques for detecting the matches if they exist.

Doan, A., et al. in [15] proposed a machine learning based system called, Learning Source Descriptions (LSD) that locates attributes matching in a semi-automatic manner. LSD needs to execute some examples of semantic mappings from the user before running on the real database to train each machine learning technique. The user needs to provide the semantic mapping for a predetermined set of data resources to be used together with the mapping to train a set of learners. However, LSD achieved a limited accuracy due to the mismatch of some tags, and also some tags need different types of learning because they are ambiguous.

The work in [16] highlighted the issue of schema matching for a relational database. A machine learning strategy based approach named Autoplex is proposed to identify the match between schema attributes exploiting data instances. Autoplex benefits from the available characteristics of database instances to determine the correspondence between a source schema and global schema. However, learners need retraining when Autoplex applied to a new domain.

A Content-Based Schema Matching Algorithm (CBSMA) adopt neural network strategy is proposed in [19]. CBSMA relies on the full discovery of data content to identify the match by analyzing the data pattern, which is conducted by training a set of neural networks. Moreover, the work introduced in [20] suggested an instance-based schema matching approach based on information theoretic discrepancy to identify the correspondences between schemas. However, the work comprises a technique that finds semantic similarity instances between compared attributes in different tables. The technique begins with extracting instances from each attribute which is going to be compared. Then, finds a set of characteristics from these instances utilizing N-gram and finally, compares the characteristics for each attribute. However, N-gram

strategy has weaknesses, because the use of N-gram to find similarity between data sources sometimes gives wrong results or even nothing, especially in cases where the instances do not have any overlap of N-gram with each other [3].

Ji, F., et al., [21] proposed new instance-based schema matching approach based on machine learning strategy. An optimal objective function is constructed as a result of the matching which determines all equivalent attributes. Experimental results of this approach elaborated that accuracy regarding precision (P) is 85%. However, the approach is suitable only for numeric instances, as the result of precision (P) dropped to 66% when string instances are considered [3]. Zaiss, K. S. [22] introduced two instance-based matching methods utilizing neural network strategy. The first method relies on the syntactic facts of the database schema to generate regular expressions or sample values that result into characterizing the concepts of ontology by their instance sets. The second method uses the instance sets to describe the contents of every instance using a set of regular expressions.

The work contributed by [23] has also highlighted the issue of syntactic and semantic schema matching in the database. They have introduced an information theoretic discrepancy based approach that aims at identifying the semantic as well as syntactic correspondences attribute via their instances sets. However, the experiment result depicts that the algorithm uses N-grams, is unable to identify the matches between attributes with string types correctly compared to the second algorithm utilizes Google similarity distance which achieved a better result for the same type of data. Besides, the work presented by [14] addressed the issue of instance based schema matching in the database. They have proposed a rule-based schema matching approach which utilizes a predefined regular expression to identify the matching patterns of instances.

Lastly, the work contributed by [8] tackled the issue of schema matching based on data instances in the relational database. He proposed a schema matching approach to identify the correspondences between attributes by fully exploiting the instances for numeric, alphabetic and mix data types. The proposed approach employs the concept of pattern recognition to create regular expression based on instances in order to identify attributes matches for numeric and mix data types. Besides, for the alphabetic data type, the approach involves Google similarity to compute the semantic similarity score to capture the semantic relationships between instances.

3. THE DEVELOPED FRAMEWORK OF INSTANCE-BASED SCHEMA MATCHING.

This section discusses the details components of instance-based schema matching framework which has been adopted from [8]. The framework aims to detect the matches between two schema attributes via their instance sets which consists of five main phases as demonstrated in Figure 1. These phases are Identifying Attributes, Classifying Attributes, Generating the Optimal Sample Size, Identify Instance Similarity and Matching Attributes, which are further explained in the following subsections.

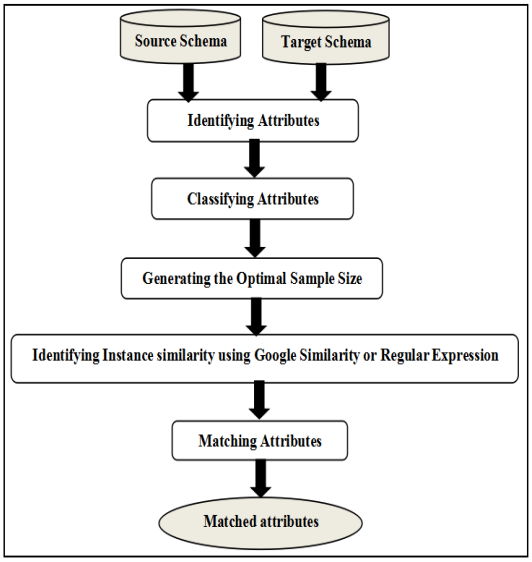


Figure 1. The phases of the instance-based schema matching framework

3.1. Identifying Attributes

This phase intends to identify the data type of each attribute of the source and the target schemas by analyzing the characters of some randomly selected instances from each attribute. Three data types of an attribute have been defined, namely: alphabetic, numeric, and mixed (string, digits and special characters). The input consists of a set of randomly generated set of instances from both source and target schemas, while the output is the identified data type of each attribute. The process starts by randomly selecting and scanning some instances of an attribute counting the number of characters for each data type. Then, compare the number of characters of the data type with the number of characters of the scanned instances. If the number of characters of the data type equivalent to the length of the instance (excluding white-spaces), and all characters are alphabetic. Then, we identify the data type of the instance as alphabetic. Similarly, if the length of the characters of the data type equals to the number of characters of the scanned instance and the characters are numeric, then, identify the data type as numeric. Otherwise, the data type of the instances is identified as a mix. Finally, the process ends by counting the number of alphabetic, numeric and mix instances and accordingly assigns an attribute to a particular data type.

3.2. Classifying Attributes

The main purpose of this phase is to reduce the number of possible comparisons needed during the matching process. This phase received the number of data types identified from the previous phase as an input to classify them into different classes based on the same derived data type. The maximum number of classes that might be introduced in this phase depends mainly on the number of data types produced from identifying attributes phase. Each class will hold several attributes having the same data type or domain. This process helps to eliminate the irrelevant comparisons between schema attributes, where attributes in each class will only be compared to each other. This step ensures that the attributes with the same data type are combined together in the same class.

3.3. Generating the Optimal Sample Size

This phase aims at extracting the optimal random sample size of instances of each attribute of the identified classes. This helps in reducing the processing time of the matching process by relying on a small portion of the instances in the database table to be used in order to determine the similarity between attributes. It is obvious that utilizing a sample of instances instead of involving the entire instances will significantly improve the performance of the matching approach, and avoid unnecessary access to a large portion of the instances. In this work, we set up the optimal sample size to be up to 50% of the actual table size to maintain a good level of accuracy [24].

3.4. Instance Similarity Identification Phase

This phase focuses on comparing attributes of different schemas belongs to the same class to check if they are representing the same entity or not. Two different instance similarity identification methods have been developed under this phase, namely: (1) Regular expression for syntactic similarity, and (2) Google for semantic similarity. Both methods attempt to identify the correspondences between attributes in each class. This phase considers the most significant phase in the instance-based schema matching process which tries to extract similarities among instances through pairwise comparisons between instance sets in order to measure the match between their attributes. Each instance is compared head-to-head (one-on-one) with each of the other instances. In this phase, we have implemented two different methods identify the similarities between instances sets. The first method regular expression relies on the syntactic similarities between instances, while the second method Google similarity employs the semantic similarities to identify the correspondences between attributes. These methods are further explained in the following subsections.

3.4.1 Regular Expression (Regexes)

Regular expression method helps in identifying the syntactic similarity between two sets of instances from two different schemas using the regular expression of the instances. A regular expression is a string containing a combination of normal characters and special characters such as (*, +, %). One of its benefits is an inexpensive process as it does not need training or learning processes. Furthermore, it is quick and concise in capturing valuable user knowledge about the domain [3, 7 - 8]. Using regular expression suggests that the set of instances should be represented as one single pattern in order to provide an accurate matching result between instances. *RegEx* is designed to find a particular regular expression that describes a set of data values (instances). Thus, it can be possible to create a regular expression that fits the majority of the instances set syntactically (formats) in order to identify the similarity between different instances sets. The process of generating a regular expression is performed in two ways regarding the data types of the attributes. For numeric attributes, the process of generating attributes *RegEx* is separately performed due to

the involvement of certain mathematical calculation, while, alphabetical and mixed attributes share the same process.

A. Generating RegEx for Numeric Data Type Attributes

Instances belong to numeric attribute consists of digits' characters only in the range of 0 - 9. Basically, regexes method needs to identify the minimum and maximum values of the attributes to generate the regular expression for a numeric attribute. The minimum and maximum values are assigned to the initial values of the attributes. In addition, the upper is also needed which is greater than the value of min and less than the value of max. Three variables need to be identified, namely: min, max, and upper. The upper is derived if one of the following conditions holds:

i. If the length of the *min* is less than the length of the *max*, then the *upper* is the *max* value based on the *min* length and not greater than the value of *max*. For example, suppose the *min* value is 654. Therefore, its length is three, the possible *max* value of the length of the *mini* value is 999. Therefore, 999 is said to be the *upper* maximum of the *mini* value length. Then we check again if the *upper* is greater than the *max* value. Therefore, the first digit of the *upper* is replaced by the first digit of the *min* value (i.e.: 699). If this new *upper* is still greater than the *max* value, the second digit of *upper* replaces the second digit of the *min* value (i.e.: 659). This iteration will subsequently perform until it meets the above condition of *upper*. However, in the case where *upper* iteration results to be equal to the *min* value, therefore the *max* value is denoted as the *upper*.

ii. When the digits' length of *min* is equal to the digits' length of *max* and *min* has at least one zero digits on the right, the *upper* is derived using the formula given below.

$$Upper = (max - (min \text{ MOD } sumz * 10)$$

B. Generating RegEx for Alphabetic and Mix Data Type Attributes

This section explains the detail steps of generating the regular expression for attributes with alphabetic and mix data using regular expression technique. The idea of generating a regular expression for alphabetic and mix data types relies on dividing an instance into a set of sub-tokens. This concept has been applied in regular expression approach to constructing a regular expression for attributes with mix and alphabetic data types. The derived sub-tokens contain a set of characters of a particular data type that will be processed separately to generate the regular expressions of the instance. Eventually, the constructed regular expressions of the sub-tokens are combined together to form the regular expression of the instance.

Where *sumz* refers to the number of zero's in the *mini*. If the value returns from the above equation less than *max* and greater than *min*, then assigned the value to *upper*. Otherwise, apply the steps in condition (i) [3]. To generate a regular expression for numeric data type attribute, an interval needs to be derived based on *min* length and its value, and the value of *upper*. The process of deriving interval and creating a regular expression for that particular interval continues until *upper* = *max*. Lastly, the created regular expressions of these derived intervals are merged together in one single regular expression using | operator to indicate the regular expression of the attribute [3].

3.4.2 Google Similarity Distance

Google similarity technique exploits the largest database which is a World Wide Web as a source of search and employs Google as a search engine for this database. The below equation describes how the Google similarity technique uses Google pages count to identify the similarity of words and phrases from World Wide Web [3, 25]:

$$\text{GSD}(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log M - \min\{\log f(x), \log f(y)\}} \quad (1)$$

Where:
f(x): refers to the number of Google hits for the search term x.
f(y): refers to the number of Google hits for the search term y.
f(x, y): represents the number of Google hits for both terms x and y together.
M: indicates the number of web pages indexed by Google.

It is very obvious that the World Wide Web is the largest available database in the world whereby millions of independent users entering the various types of information. The idea of using Google similarity distance technique involves this database to help in producing automatic semantics of useful quality of relationships between targeted subjects [3, 8, 25]. In contrast to other semantic methods such as Latent semantic analysis (LSA), WorldNet and thesaurus that uses a closed collection of the limited size of documents. Google similarity technique works as follows: To identify the semantic relationship between two different terms, for example (doctor and professor) representing two different instances for different attributes. We first start searching in Google web pages for each term separately to find the number of occurrences of these terms in Google web pages. Then, we continue the search for those pages contain both terms "doctor" and "professor" together to retrieve the total number of pages where these two are found. Eventually, we will have the number of hits for both founded terms, and the number of hits for each term found separately. Furthermore, we also involve the current total number of pages indexed by Google engine in WWW database which is 3000,000,000 approximately. Then substitute the obtained values in the equation (1) to produce the similarity degree GSD between the two terms "doctor" and "professor". When the value of GSD is close to zero, indicates that no semantic relationship between the two terms is detected. Otherwise, if the score value is close to 1, then it is assumed that the two terms are semantically related, and the two values represent a property of the same entity [26].

A. Find Similarity Score for Attributes

Google similarity is the second approach that has been considered in this thesis to determine the correspondence between attributes. It is used to identify the match between alphabetic, numeric, and mixed data type attributes. The idea of Google similarity approach is relying on computing the semantic similarity score between instances to discover the semantic relationship between attributes of the source and target schemas. It is in contrast to regular expression approach that utilizes the schema information without taking into account the implicit semantic relationship between attributes.

3.5. Attribute Matching Phase

Attribute matching is the last stage in the process of instance-based schema matching. In this phase, we attempt to identify the correct match between the attributes that shared same data type and eventually mapping them. The process is carried out after performing the task of syntactic and semantic matching in the previous phase. In this phase, a decision needs to be made whether two different attributes are considered similar or not. Due to considering two different techniques which are a regular expression and Google similarity to identify the match between attributes; consequently, in this phase, two matching mechanisms have been implemented to handle the mapping task, namely: regular expression-based attribute matching and Google similarity based attribute matching.

4. EXPERIMENT RESULTS

To fairly evaluate the instance-based schema matching techniques considered in this paper, two different types of the data sets have been used in the experiment study, namely: synthetic and real data sets. For synthetic data set, an online data generator named BETA has been used. In this type of data set, the attributes are generated by setting out their appropriate names, data types, data ranges (if needed), and the size of the data. We have developed a university database that consists of a set of attributes with different types of data and varying range of values. The main reason behind selecting this type of data set is to obtain a deep insight and better understanding of the effect of data characteristics on the behavior and the performance of the developed under comparison. Furthermore, two real data sets (Restaurant and Census) have been used in the experiments to examine fairly the approaches considered in this thesis. These real data sets have been used in most previous works related to the area of schema matching in database, and particularly for instance-based schema matching [8- 9, 14, 27- 28]. Both Restaurant and census data sets are available online.

In the experiment two sub-tables have been derived from the original tables of the data sets. These two sub-tables represent the source schema and target schema in the experiments. The set of attributes belongs to the source and target schema has been generated randomly and the number of attributes in each sub-table is equivalent to the number of attributes of the original table. For each sub-table, a set of random different instances is inserted referring to the original table of the data set [8, 29]. Two analyses that have been conducted, the first analysis emphasizes on identifying the optimal sample size of instances to achieve acceptable accuracy results for the matching process. The second analysis intends to compare the performance of both techniques in terms of precision (P) and recall (R) and F-measure (F).

4.1 Experiment 1

This analysis highlights the experiment of selecting the optimal sample size of tuples to be used during schema matching process. The process of sample size selection is performed by *generating the optimal sample size* phase of instance-based schema matching. In this analysis, we attempt to study the impact of the sample size of the tuples on the quality of the matching result in terms of precision (*P*), recall (*R*), and F-measure (*F*) for both strategies. The sample size is among the important parameters that influence the quality and the performance of the matching process [3, 8, 24]. Therefore, discovering the best sample size of instances is extremely needed in order to measure the accuracy of the considered techniques. We start from 10%, and the sample size gradually increased by 10% in the subsequent experiments up to 50% of the actual table size. This increment helps to discover whether the approaches that have been considered require a large number of instances in order to achieve an accurate match between schemas. From this analysis, it has been explored that increasing the sample size leads to a better result of Precision (*P*), Recall (*R*), and *F*-measure (*F*) for both approaches. Table 1 demonstrates the sample size considered in each experiment. All these experiments used the same data set and ended up when sample size reached 50%. Each experiment has been executed five times measuring the *P*, *R*, and *F* and averaged these results.

Table 1. Sample size for each experiment

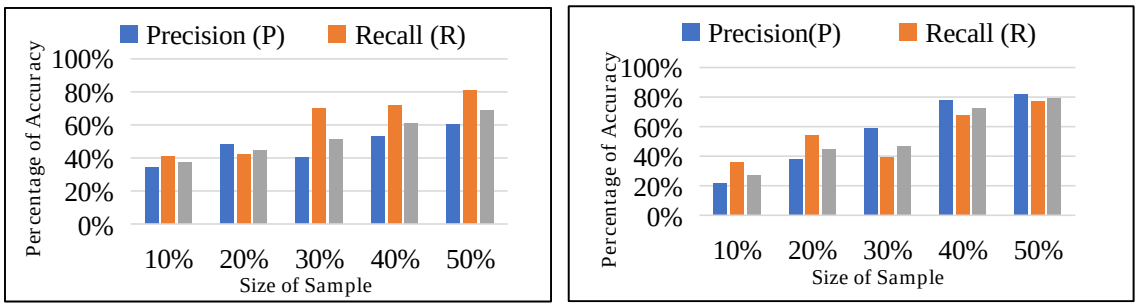
Experiment	Size of Samples
Experiment 1-1	10%
Experiment 1-2	20%
Experiment 1-3	30%
Experiment 1-4	40%
Experiment 1-5	50%

4.1.1 Result of Experiment 1

This sub-section presents the detail results of Analysis 1. In this analysis, various experiments have been conducted on two real-world data sets (i) Restaurant data set and (ii) Census data set, and one synthetic data set (i) University data set to identify the optimal sample size for the best matching result.

4.1.1.1 Result of Experiment 1 Related to Restaurant Data set

In this analysis, a real world data set related to Restaurant domain is used to determine the optimal sample size to be used in both approaches (Regular expression and Google similarity). Restaurant data set consists of a list of restaurants in two popular websites, namely: Zagat and Feodor. The data set comprises of five attributes contain instances representing two different data types *alphabetic* and *special characters (mixed)*. Selecting the optimal sample size has a significant impact on reducing the number of comparisons between instances, which further reduce the processing time of the matching process. Figure 2(a) and 2(b) demonstrate the results of Precision (*P*), Recall (*R*) and F-measure (*F*) for the experiments of analysis 1 for both methods Regular expression and Google similarity respectively. It is very clear that the accuracy of the matching result using regular expression strategy increases when the sample size increase as shown in Figure 2. Notice that when the sample size is 50% the percentages are 60% and 81% for precision (*P*) and recall (*R*) respectively. However, in Figure 2(b) for Google similarity technique, the percentages of precision (*P*) and recall (*R*) has increased up to 82% and 77% respectively.

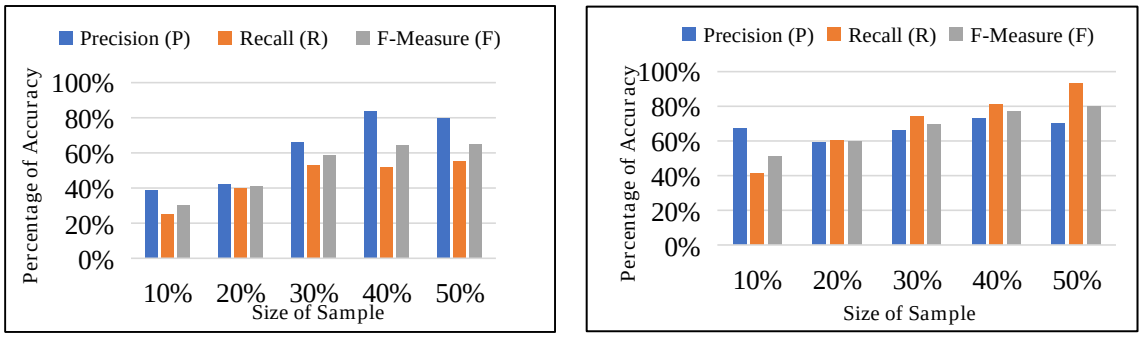


(a) Regular Expression (b) Google Similarity

Figur 2. Results of Precision (*P*), Recall (*R*) and F-measure (*F*)

4.1.1.2 Result of Experiment 1 Related to Census Data set

The Census real data set contains weighted census data extracted by Barry Becker in 1994 from the Census database, to determine the optimal sample size that would result in reducing the number of comparisons between instances to identify the instance similarity, which further reduces the processing time of the matching process. The instance sets of this data set involve the three data types, which are a numeric, alphabetic and special character. Figure 3(a) and 3(b) demonstrate the results of precision (*P*), recall (*R*) and *F*-measure (*F*) for this analysis on Census data set using Regular expression and Google similarity respectively. It can be noticed that for regular expression technique, utilizing the large size of instances sample can considerably improve the accuracy of the matching results. Similarly, the accuracy of the matching results involving Google similarity has been improved when the sample size increased as shown in Figure 3(b). In Figure 3(a), the percentage of the recall (*R*) slightly increased to 55%, nevertheless regular expression substantially improved the percentage of the precision (*P*) and *F*-measure (*F*) simultaneously from 39% and 25% to 80% and 55% respectively when the sample size has increased. Lastly, in Figure 3(b), the percentages of the precision (*P*) and Recall (*R*) are slightly improved when the sample size increased. Meanwhile, the best result achieved by Google Similarity was approximately 80% for F-measure with only 50% size of instance sample. This indicates that Google Similarity technique has the capability to discover the matching between attributes precisely without paying much consideration to a number of instances.



(a) Regular Expression (b) Google Similarity

Figure 3. Percentage of *P*, *R*, and *F* for Census data set

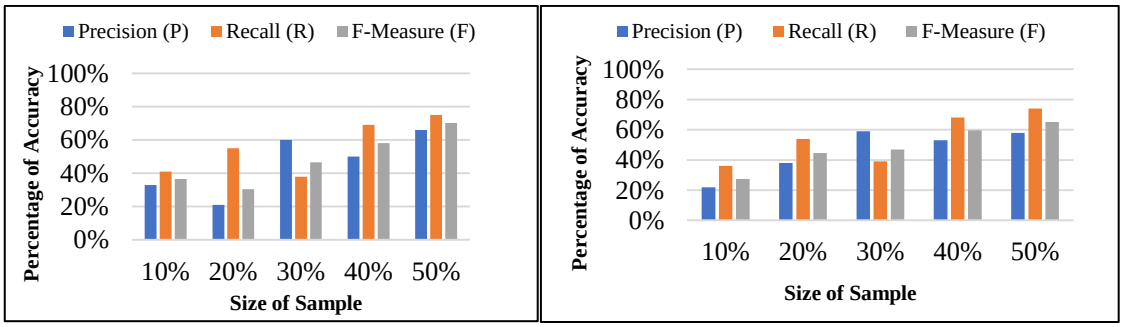
4.1.1.3 Result of Experiment 1 Related to Synthetic Data set

In this section, we highlight the analysis results on the synthetic data set that has been generated to evaluate the performance of the instance-based schema matching process. Figure 4(a) presents the results for synthetic data set using regular expression technique by varying the sample size of instances in the range of 10%-50%. It is clear that in all cases the percentages of precision (*P*), recall (*R*) and F-measure (*F*) increases when the sample size increases. Hence, it can be concluded that the best optimal sample size that achieved a most accurate result in terms of precision is 50% of the actual table size representing the number of tuples that will be involved in the process of instance-based schema matching.

Figure 4(b) depicts of this analysis. The best result achieved for precision using Google similarity was 58%. Lastly, from the obtained results of both figures, we noticed that Google similarity outperforms regular expression in terms of precision (*P*). This is because Google similarity relies on the semantic aspect on data instances when identifying the correspondence between attributes. In contrast, regular expression achieved a higher percentage for recall (*R*) compared with Google similarity. This is because regular expression identifies the matching between instances based on the syntactic similarity between instances and there is a large number of attributes with numeric and mix data types in this data set.

4.2 Experiment 2

Analysis 2 concentrates on examining and comparing the performance of both matching techniques that considered in this research work. The parameter setting of this analysis in terms of sample size has been set to 50% of the actual table which has been identified as the optimal sample size. The results reported in this section comprises of the three different data sets involved in this study, namely: Restaurant, and Census, and synthetic data sets.



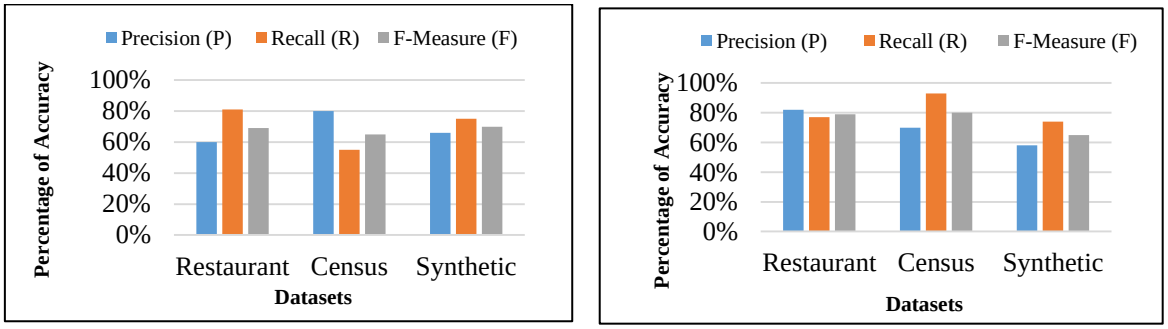
(a) Regular Expression (b) Google Similarity

Figure 4. Percentage of *P*, *R*, and *F* for synthetic data set

4.2.1 Result of Experiment 2

Figure 5(a) and 5(b) present the percentage of the accuracy result of instance-based schema matching on Restaurant, Census, and synthetic data sets using Regular expression and Google similarity respectively. Figure 5(a) describes the accuracy result of the instance-based schema matching process using Regular expression strategy. From the figure, we noticed that Regular expression technique achieved the highest accuracy on Census data set with up to 80% in terms of precision (*P*). This is due to the characteristic of Census data set which comprises of four attributes with the numeric data type and seven attributes with the alphabetic data type. Also, it can be concluded that the highest accuracy in terms of recall (*R*) using Regular expression has been achieved on Restaurant data set. This is because Restaurant data set consists of three attributes with the alphabetic data type and two attributes with mix data type. Lastly, regular expression achieved a better result on the synthetic data set compared with restaurant data set. However, the percentage of recall (*R*) on Restaurant data set is slightly better compared with the percentage of the recall (R) on the synthetic data set.

Figure 5(b) demonstrates the results of Restaurant, Census and synthetic data sets using Google similarity. From the results, it is obvious that Google similarity achieved the highest accuracy result in terms of precision (*P*) on Restaurant data set. While the best accuracy results achieved in terms of recall (*R*) and *F*-measure are on Census data set with 93% and 80% respectively. Besides, Google similarity has achieved a slightly better result in terms of precision (*P*) and *F*-measure on synthetic data set compared with restaurant data set. Nevertheless, the percentage of recall (*R*) is higher on restaurant data set.



(a) Regular Expression (b) Google Similarity

Figure 5. Matching Results using Regular Expression

5. DISCUSION

From the results that have been reported throughout this paper, we can conclude that the both matching methods achieved good results. Besides, both methods also introduced an acceptable range of time to discover the matching between attributes in different schemas. Additionally, it can also be observed that Google similarity is more appropriate to handle similarity between instances contain *alphabetic* data type compared to Regular expression. However, Regular expression method is more suitable for handling similarity between instances contains *numeric* and *mix* data types. We can also notice that the sample size of

the data instances has also influenced on the quality of the matching results in which the percentage of the accuracy increases significantly when the sample size increases. This can be seen clearly for regular expression method where the sample size of instances can significantly impact on the accuracy results. This may require a considerable amount of instances to avoid miss representation of the attribute's pattern during the process of constructing the regular expression. Similarly, for Google similarity, the amount of data instances has a great influence on the processing time and the quality of the match. Although the matching can be performed when few instances can be found in the database attributes by calculating the average of similarity scores. Nevertheless, a larger number of instances can either positively or negatively impacts on the average scores which subsequently inspired the matching quality. While, for time optimization, Google similarity is actually proportional to the sample size of tuples. When a large amount of instances used, the processing time would be longer and vice versa. This is due to the fact that Google similarity relies mainly on the internet, and involve Google search engine to accomplish the matching process. Hence, internet speed can directly affect the processing time. Furthermore, it can also be observed that Google similarity depends on the number of hits of a specific term. For example, if the term is unclear, then this would result in a low number of hits compared to the number of pages indexed by Google. Therefore, this leads to reduce the similarity score which further declines the matching accuracy.

6. ACKNOWLEDGEMENTS

In this paper we have conducted an empirical comparative study between two different instance-based schema matching techniques, namely: Google similarity and regular expression. The study sought to compare the two techniques with several synthetic and real data sets. It can conclude that regular expression technique is not suitable to be used to handle instances of attributes with string data types. However, the approach is very effective and outperforms Google similarity for attributes with mix and numeric data types instances. Similarly, Google similarity seems to be appropriate for attributes with alphabetic data type extracting the semantic relationship between the instance sets. Nevertheless, it is inappropriate to be utilized for schema attributes contain mix and numeric data. We also conclude that regular expression relies mainly on a sample size of instances to achieve high accuracy. The accuracy of the matching result increased when the sample size is large.

REFERENCES

[1] Lenzerini, M., "Data integration: A theoretical perspective". In*: Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. 2 – 6- June- 2002, Madison, Wisconsin, USA, 233-246.

[2] Do, H., "Schema Matching and Mapping-based Data Integration: Architecture, Approaches, and Evaluation". VDM Verlag Saarbrücken, Germany, 2007.

[3] Osama, A. Mehdi., "A New approach for Instance based-schema matching". Unpublished Master Dissertation. Universiti Putra Malaysia, Kuala Lumpur, Malaysia, 2014.

[4] Bernstein, P. A., Madhavan, J., Rahm, E. "Generic schema matching, ten years later". In*: Proceedings of the 37th International Conference on Very Large Data Bases*, August 29th - September 3rd 2011, Seattle, Washington, USA, 695-701.

[5] Tian, A., Kejriwal, M., Miranker, D.P., "Schema matching over relations, attributes, and data values". In: *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*. 30 - June – 2 July – 2014, Aalborg, Denmark.

[6] Gozudeli, Y., Karacan, H., Yildiz, O., Baker, M., Minnet, A., Kalender, M., Akcayol, M., "A New method based on tree simplification and schema matching for automatic web result extraction and matching". In*: Proceedings of the International Multi Conference of Engineers and Computer Scientists*. 18- 20 -March – 2015, Hong Kong, China, 1-5.

[7] Jain, S., Tanwani, S., "Schema matching technique for a heterogeneous web database". In*: Proceedings of the 4th International Conference on the Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*. 2 – 4- September- 2015, Noida, India, 1-6.

[8] Osama, A. M., Hamidah, I., Lilly S. A., "An Approach for Instance Based Schema Matching with Google Similarity and Regular Expression". *The Int. Arab J. of Info. Tech*., 2017. No. 5.

[9] Munir, S., Khan, F., Riaz, M.A. "An instance-based schema matching between opaque database schemas". In: *Proceedings of the 4th International Conference on Engineering Technology and Technopreneuship (ICE2T)*. 27-29- August- 2014, Kuala Lumpur, Malaysia, 177-182.

[10] De Carvalho, M.G., Laender, A.H., GonçAlves, M.A., Da Silva, A.S., "An evolutionary approach to complex schema matching". *J. of Info. Sys*, 2013, 38(3), 302-316.

[11] Osama A. Mehdi, Hamidah, I., Lilly S.A., "Instance based matching using regular expression". *Procedia Com. Sci*., 2012, 10, 688-695.

[12] Zhao, H., Ram, S., "Combining schema and instance information for integrating heterogeneous data sources". *J. of Data & Know. Eng.*, 2007, 61(2), 281-303.

[13] Leme, L.A.P.P., Casanova, M.A., Breitman, K.K., Furtado, A. L., "Instance-Based OWL Schema Matching". In: *Proceedings of the 11th International Conference on Enterprise Information Systems, (ICEIS 2009)*, 6- 10- May – 2009, Milan, Italy, 14 - 26.
[14] Zapilko, B., Zloch, M., Schaible, J. "Utilizing regular expressions for instance-based schema matching". In: *Proceedings of the 7th International Conference on Ontology Matching*. 11- November – 2012, Boston, MA, USA, 240-241.
[15] Doan, A., Domingos, P., Halevy, A.Y., "Reconciling schemas of disparate data sources: A machine-learning approach". In: *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, 2001 May 21-24, Santa Barbara, California, USA, 509-520.
[16] Berlin, J., Motro, A., "Database schema matching using machine learning with feature selection". In: *Proceedings of the 14th International Conference on Advanced Information Systems Engineering (CAiSE'02)*. May 27-31, 2002, Toronto, Ontario, Canada, 452-466.
[17] You, L., Dong-Bo, L., Wei-Ming, Z., "Schema matching using neural network". In: *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, 19- 22- September- 2005, France, 743-746.
[18] Bilke, A., Naumann, F., "Schema matching using duplicates". In: *Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, April 05 - 08, 2005, Tokyo, Japan, 69-80.
[19] Yuan, Y., Mengdong, C., Bin, G., "An effective content-based schema matching algorithm". In: *Proceedings of the 2008 International Seminar on Future Information Technology and Management Engineering (FITME)*, 20- November- 2008, 7-11.
[20] Dai, B. T., Koudas, N., Srivastava, D., Tung, A.K., Venkatasubramanian, S., "Validating multi-column schema matchings by type". In: *Proceedings of the 24th International Conference on Data Engineering (ICDE2008)*. Cancun, Mexico, 120-129.
[21] Ji, F., Xiaoguang, H., Yuanbo, Q., "An instance-based schema matching method with attributes ranking and classification". In: *Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery FSKD'09*. 14-16 Aug. 2009, Tianjin, China, 522-526.
[22] Zaiss, K. S., "Instance-based ontology matching and the evaluation of matching systems". Unpublished doctoral Dissertation. University of Dusseldorf, 2010, Germany.
[23] Partyka, J., Parveen, P., Khan, L., Thuraisingham, B., Shekhar, S., "Enhanced geographically typed semantic schema matching". *J. of Web Semantics: Sci., Serv. & Agents on the World Wide Web*., 2011, 9(1), 52-70.
[24] Köhler, H., Zhou, X., Sadiq, S., Shu, Y., Taylor, K., "Sampling dirty data for matching attributes". In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*. 6 – 11- June – 2010, Indianapolis, Indiana, USA, 233-246.
[25] Cilibrasi, R.L., Vitanyi, P., "The Google similarity distance". *IEEE Tran. on Know. & Data Eng., J., 2007*, 19(3), 370 - 383.
[26] Rahm, E., Bernstein, P. *On matching schemas automatically*. The VLDB J., 2001, 10(4), 334-350.
[27] Preethi, M., Madhumitha, R., "Ontology based approach for instance matching". *Int. J. of Sci. & Eng. Research*, 2014, 5(3), 12-17.
[28] Ferragut, E.M., Laska, J. "Nonparametric Bayesian Modeling for Automated Database Schema Matching". (Technical Report) New York, Cornell University online library: Cornell University, 2015.
[29] Jaewoo, K., Naughton, J.F. "Schema matching using interattribute dependencies". *IEEE Tran. on Know. & Data Eng.*, 2008, 20(10), 1393-1407.

BIOGRAPHIES OF AUTHORS

Mogahed Alzeber received his master degree in Information Technology from International Islamic University Malaysia (IIUM), in 2015. 2016. His current research interests include data integration, schema matching in database and query processing.

Ali A. Awan is currently an assistant professor at Kulliyyah (Faculty) of Information and Communication Technology, International Islamic University Malaysia (IIUM), Malaysia. He received his Master of Computer Science (2009) and Ph.D in Computer Science (2013) from Universiti Putra Malaysia (UPM), Malaysia. His research interests include preference queries, skyline queries, probabilistic and uncertain databases, query processing and optimization and management of incomplete data, data integration, location-based social networks (LBSN), recommendation systems, and data management in cloud computing.

Azlin Nordin received her PhD in Computer Science from the University of Manchester in 2013. She is currently an academic at the Department of Computer Science, Kulliyyah of Information and Communication Technology at International Islamic University Malaysia. Her research interest is on requirements reuse, requirements patterns, and requirements validation.

Abedallah Zaid Abualkishik. is currently an assistant professor at the College of Computer Science and Information Technology. He has earned a P.h.D and master's degree in Software Engineering from Universiti Putra Malaysa, (UPM), Malaysia. His current research interests are in the areas of software metrics conversion, empirical software engineering and software quality.