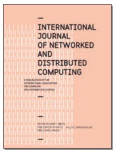
ATLANTIS PRESS

International Journal of Networked and Distributed Computing
Vol. **7**(**3**); July (2019), *pp.* 100–106
DOI: https://doi.org/10.2991/ijndc.k.190710.001; ISSN 2211-7938; eISSN 2211-7946
https://www.atlantis-press.com/journals/ijndc



Research Article

Semantic Schema Matching for String Attribute with Word Vectors and its Evaluation

Kenji Nozaki[1,*], Teruhisa Hochin[2], Hiroki Nomiya[2]

[1]*Graduate School of Information Science, Kyoto Institute of Technology, Matsugasaki, Sakyo-ku, Kyoto 606-8585, Japan*
[2]*Faculty of Information and Human Sciences, Kyoto Institute of Technology, Matsugasaki, Sakyo-ku, Kyoto 606-8585, Japan*

ARTICLE INFO

Article History

Received 07 February 2019
Accepted 06 May 2019

Keywords

Instance-based schema matching
schema matching
semantic matching
Word2Vec

Abstract

Instance-based schema matching is to determine the correspondences between heterogeneous databases by comparing instances. Heterogeneous databases consist of an enormous number of tables containing various attributes, causing the data heterogeneity. In such cases, it is effective to consider semantic information. In this paper, we propose the instance-based schema matching considering attributes' semantics. We used Word2Vec to match attributes of character strings. The result shows a possibility to detect matching between attributes with high semantic similarity.

1. INTRODUCTION

Databases are independently developed according to the purpose of use. Therefore, data heterogeneity occurs even in the databases expressing the same entity. This heterogeneity increases the data processing complexity and raises the need for data integration. However, it is difficult to integrate and manage databases independently developed. This is because data expressions and design are different. Identification of correspondence between schemas is an important issue in data integration. Therefore, many schema matching processes have been proposed to find the correspondence between schemas and integrating them [1].

To apply schema matching is inappropriate when schema does not use a unified standard. Furthermore, it is not so effective to use schema design in formation and attribute names directly. Even if it shows the same entity, different abbreviations and expressions may be used. In such cases, it is difficult to determine the correspondence between attributes using schema matching. When the schema information is not available or insufficient to use for the schema matching, finding the correspondence of instances is an alternative approach to the schema matching. Instances contain the exact features of the actual content of the attribute. Using instances makes it possible to find correspondence between attributes, even when the schema information is inaccurate.

The instance-based schema matching methods analyze instances syntactically and semantically for the majority, and determine correspondence between attributes [2]. Syntactic methods include N-grams and regular expression, while semantics methods include Latent Semantic Analysis (LSA), WordNet, thesaurus and Google similarity. The N-gram is a model that has been used for spelling correction, word breaking and text summarization. The analysis process involves dividing a word or text into consecutive tokens and obtains a set of fragments. Similarity between words or texts can be achieved by comparing sets of fragments obtained from the N-grams. The regular expression analyzes patterns of text. Values in the attribute are analyzed for the occurrence pattern of the characters, and compared with the value of other attribute for the matching between the attributes. The Google similarity is used in the World Wide Web (WWW) containing a large amount of online pages. In this method, WWW is considered as a real semantic database because the context information has been entered by millions of independent users. The Google similarity calculates the semantic similarity score for attributes using the result of searching their instances, and determines the correspondence of attributes. Using such methods, instance-based schema matching has been challenged.

In this paper, we used the Word2Vec [3–5] for a semantic comparison [6]. The Word2Vec is a neural network that performs text processing. It can vectorize words in hundreds of dimensions and perform addition and subtraction between words. For example, we can calculate words like "king − man + woman = queen." These vectors reflect the semantic relations of each word. Hence, we vectorized instances using Word2Vec, and made a semantic comparison of instances. We aimed to find the correspondences between attributes from the similarity scores and investigated whether it is effective for instance-based schema matching. Here, the notion of "instance" refers to the values of schema attributes, and the notion of "attribute" refers to the column of a table in a database.

This paper is organized as follows. In Section 2, we describe related works. Section 3 presents an instance-based schema matching method applicable to the character string attributes. Section 4 describes our

*Corresponding author. Email: ken23mybgbc2@gmail.com

experiments with additional datasets for demonstrating the benefits of the proposed method. We discuss about the results in Section 5. Finally, Section 6 draws the conclusion and points out the future work directions.

2. RELATED WORKS

2.1. Schema Matching

Zhao and Ram [7] proposed a cluster analysis approach to semi-automate the Interschema Relationship Identification (IRI) process. IRI process determines the relationships between objects in heterogeneous database schemas. IRI is the classical schema integration problem when generating integrated data source. This paper classified interschema relationships using various clustering methods including *k*-means, hierarchical clustering and self-organizing map. Furthermore, they developed a prototype system that visualized the clustering result, and discussed the importance of the visualization for user evaluation. Input features included the name similarity (attribute, entity and relation) and schematic information. As a result, Zhao and Ram described that direct semantic features such as attribute name similarity are more important than indirect semantic features. However, comparison between attribute names is not so effective in real-world heterogeneous databases due to differences of data expression. In such cases, the quality of the semantic clustering for integration of attribute degenerated seriously.

A semantic integration algorithm is proposed by Partyka et al. [8] that is called TSim using Normalized Google Distance (NGD), and the problem of N-gram used in instance-based schema matching was raised. A popular method is accomplished by extracting instance values from the compared attributes, extracting a characteristic set of N-grams from these instances, and finally comparing the respective N-grams for each attribute. N-gram similarity is based on a comparison of the concepts of entropy and conditional entropy known as Entropy-Based Distribution (EBD). However, this method is weak against few shared instances, and the accuracy is significantly low. In the proposed algorithm, individual keywords are extracted from the compared attributes and the type is determined by grouping keywords of the same type using K-medoid clustering based on semantic distance metrics called NGD. The EBD is calculated by comparing all instances of keywords representing each type. Comparing these two methods, the proposed TSim algorithm was able to determine the exact correspondence. This is because the semantic comparison using NGD made it possible to eliminate the syntactic dependence of the instances.

Mehdi et al. [9] proposed an instance-based schema matching method using the Google similarity and the regular expression. They divided attributes into three classes by analyzing the characters of an instance. Attributes are classified as numeric, alphabetic and mix data types whose instances consist of numeric, alphabet and symbols. The regular expression analysis is applied to numeric and mix data types and utilized to analyze the character appearance pattern of instances for syntactic similarity. The Google similarity is applied to alphabetic data type and utilized to calculate the similarity of pair of attributes for semantic similarity. The Google similarity uses WWW that is considered as a large semantic database entered by millions of users. Furthermore, they extracted a sample of instances for each attribute of the class based on the optimal sample size. This is due to the reduction of the processing time. As a result, their proposed approach could identify 1–1 matches with high accuracy despite using not entire instance but sampling.

2.2. Word2Vec

The Word2Vec [3–5] is a neural network composed of two layers and performs text processing. It is a method of analyzing large amount of text data and vectorizing the meaning of words in hundreds of dimensions. In order to obtain the vector expression of words, it is considered to solve a task predicting a certain word from surrounding words. The Word2Vec has two learning methods. One is called Skip-gram, and the other is called Continuous Bag-of-Words. The former model predicts the surrounding words from an inputted word, and the latter model predicts a certain word from surrounded words. In both models, the weight matrix connecting the input layer and the hidden layer is the vector expressions of the word generated by the Word2Vec.

Using the vector expression of words, it is possible to calculate the similarity and perform addition and subtraction between words. For example, the result of inputting “king” is shown in Table 1. Cosine similarity is used for the calculation of similarity between vectors of words. The similarity value is −1 to 2.

This result shows that the words similar to “king” are obtained. Furthermore, the result of “king – man + woman” is shown in Table 2.

The “queen” is at the top, and Table 2 has a word “princess” which is not in Table 1. In addition, the similarity of “kings” in Table 2 is much lower than the similarity in Table 1. Thus, it may be considered that the semantic of the calculation is reflected.

Table 1 | Result of “king”

Word	Similarity
Kings	0.7138
Queen	0.6511
Monarch	0.6413
Crown prince	0.6204
Prince	0.616
Sultan	0.5865
Ruler	0.5798
Princes	0.5647
Prince Paras	0.5433
Throne	0.5422

Table 2 | Result of “king – man + woman”

Word	Similarity
Queen	0.7118
Monarch	0.619
Princess	0.5902
Crown prince	0.5499
Prince	0.5377
Kings	0.5237
Queen consort	0.5236
Queens	0.5181
Sultan	0.5099
Monarchy	0.5087

In this research, we use the Word2Vec for vectorizing instances and calculate the vectors for the attribute matching. By calculating the cosine similarity from these vectors generated from the Word2Vec, the similarity between the attributes is calculated and correspondences is determined. Since the Word2Vec is a neural network and conducts leaning of text data, it is easy to reproduce the same situation unlike the Google similarity. The Google similarity uses WWW which is rewritten to many users so the execution results change according to the execution timing. In other words, the data source is considered to be unstable and unclear. This difference is a better point of the Word2Vec. In our experiments, we did not conduct training a model for our experiments newly because the required model only needs to involve the vector expression that composed general semantics of words. We used pre-trained model that is published on the Internet [10].

3. PROPOSED METHOD

In this section, we propose a method that is an instance-based schema matching approach applicable to the character string attributes. The input of this method is two tables in the schemas for the attribute matching, and the output is the pairs of attributes regarded as corresponded. Our target attributes are composed by instances consisting only of character strings. The proposed method uses Word2Vec to get the semantics of words and character strings included in instances. It leads semantics of attributes and makes it possible to detect corresponding schemas attributes.

First, it is necessary to retrieve the target attribute. The input tables are decomposed into attributes, and we get attributes having instances only composed of character strings from the source schema and the target schema. Second, we retrieve the type of instances in character string attributes. For each instance, a vector of an instance is calculated using the Word2Vec as follows:

(i) Decompose an instance word-by-word.

(ii) Calculate the sum of vectors of these words using Word2Vec.

(iii) Ignore the word whose vector cannot be calculated.

(iv) If all words included in the instance are ignored, the vector of the instance is equal to zero.

Third, an attribute vector is calculated as shown in the following Equation (1) using instance vectors included in the attribute.

$$v_a = \sum v_i * \frac{K_i}{N_a} \tag{1}$$

Here, v_i is a vector of an instance included in the attribute, K_i is the number of instances of v_i, and N_a is the total number of instances. This vector of attribute v_a is calculated by considering the appearing ratio of values included in the attribute.

Finally, the cosine similarity is calculated between the vectors of the attribute of the source schema and that of the target schema. At this time, we set the threshold, and the pairs of the attributes are regarded as matched when the similarity values exceed the threshold. As a result, pairs of attributes with higher similarity are regarded as matching targets, and output as a list.

For example, one table decomposed into some attributes, and the attribute marital-status is retrieved. It includes four values that are “Married-civ-spouse”, “Divorced”, “Never-married” and “Separated.” To get the vector of the marital-status, the values should be decomposed and calculated. Taking “Married-civ-spouse” as an example, it is decomposed into three words that are “married”, “civ” and “spouse”. These words are vectorized by the Word2Vec one-by-one; “married”, “civ” and “spouse” are [0.018, −0.117], [0.129, 0.035] and [0.060, −0.398], respectively. The sum of the vectors is calculated and making it the vector of the value; the vector of the instance “Married-civ-spouse” is calculated as [0.207, −0.480]. Other values are calculated in the same method, and we get the vectors of all values; “Divorced”, “Never-Married” and “Separated” are [−0.001, −0.202], [0.042, −0.163] and [−0.328, −0.113], respectively. The vector of the attribute marital-status is calculated using the vectors of values and the number of values with the above equation. Here, the numbers of values concluded in “Married-civ-spouse”, “Divorced”, “Never-married” and “Separated” are 50, 30, 15 and 5, respectively. As a result, the vector of the attribute marital-status is calculated as [0.0904, −0.3307]. This is a calculation example of the proposed method about one attribute and this method is used for some attributes in two tables. The vectors of the attributes are compared between each table using the cosine similarity, and the pairs of the attributes whose similarity values are higher than the threshold are determined as matched.

4. EXPERIMENTS

We conducted some experiments to confirm the proposed approach as follows. Subsection 4.1 described about the real datasets we used. In Subsection 4.2, an experiment was conducted to obtain the threshold that determined whether attributes were matched or not. The threshold was determined from the value of non-matched pairs that were calculated by comparing the same dataset. In Subsections 4.3 and 4.4, we conducted the experiments that were the comparison of two similar datasets. These experiments verified how effective the semantic comparison of instances. In Subsection 4.5, we conducted to examine whether it was possible to find pairs of attributes indicating the same semantics between the completely different datasets. This experiment verified whether it was possible to find the pairs of attributes with same semantics by the semantic comparison of instances.

4.1. Datasets

We used five real datasets that have been used in an experimental study, namely: Adult, Census-Income, Bank, IBM Human Resources (HR) Analytics Employee Attrition & Performance (IBM HR) and HR Data Set [11,12]. The Adult dataset is a census dataset and contains 32,561 records and 11 attributes. Seven attributes have character string instances, which are workclass, education, relationship, race, sex, marital-status and native-country. The details of the Adult are shown in Table 3. The Census-Income dataset contains 199,523 records and 40 attributes. Thirty-three attributes have character string instances, and some attributes have the same entity as the seven attributes of the Adult dataset. The details of these attributes are shown in Table 4. Although these two datasets are divided into 2/3 and 1/3 for training and test datasets, the above description refers to training datasets and these datasets were used for experiments. The Bank dataset contains 45,211 records and 16 attributes. Six attributes have character string instances. Three attributes of them have the possibility of matching with the attributes of the Adult dataset, which are job, marital and education. The details of them are described in Table 5. The IBM HR dataset contains 1470 records and 35 attributes. It contains six attributes that are regarded to have character string instances and some of them are shown in Table 6. The HR Data Set contains 302 records and 21 attributes. It contains 12 attributes that are regarded to have character string instances and some of them are shown in Table 7.

We compared these five datasets to find correspondence of the attributes that express the same entity between them.

In the experiments, we used the pre-trained model that Google published [10]. It was trained on part of Google News dataset (about 100 billion words). This model contains 300 dimensional vectors for 3 million words and phrases.

4.2. Decision of the Threshold

4.2.1. Experiment

First of all, we experimentally obtained the threshold of similarity score that determines whether the pair of attributes was matched or not. We examined similarity score between non-similar attributes by comparing each attribute in the Adult dataset. The threshold of similar or non-similar was decided from the result.

Table 3 | Details of adult

Attribute	Type of value contained	Number
Workclass	Private, Self-emp-inc, Federal government, Never-worked, ...	8
Education	Bachelors, Some-college, 11th, HS-grade, Masters, ...	16
Marital-status	Married-civ-spouse, Divorced, Never-married, ...	7
Occupation	Tech-support, Craft-repair, Other-service, Sales, ...	14
Relationship	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried	6
Race	White, Black, Asian-Pac-Islander, ...	5
Sex	Female, Male	2
Native-country	United-States, Cambodia, England, Canada, ...	41

4.2.2. Result

Table 8 shows similarity of comparing attributes in the Adult dataset. The similarities between the same attributes were 1.00. Most of the similarities between non-similar attributes were 0.2–0.5 or less. However, the similarity between marital-status and relationship exceeded 0.6, which was higher than similarities of other pair of non-match attributes. The marital-status indicates marriage information

Table 5 | Details of bank

Attribute	Type of value contained	Number
Job	Admin, unemployed, management, housemaid, ...	12
Marital	Married, divorced, single	3
Education	Unknown, secondary, primary, tertiary	4

Table 6 | Details of analytics employee attrition & performance

Attribute	Type of value contained	Number
Department	Sales, Research, Development, Human Resources	3
Gender	Female, Male	2
Job role	Sales Executive, Research Scientist, Manufacturing Director, Healthcare Representative, ...	9
Marital status	Single, Married, Divorced	3

Table 7 | Details of human resources data set

Attribute	Type of value contained	Number
Sex	Female, Male	3
MaritalDesc	Married, Divorced, Single, ...	5
Department	Admin Offices, Executive Office, IT/IS, ...	7
Position	Software Engineer, Administrative Assistant, Database Administrator, Production Manager, ...	24

Table 4 | Details of census-income

Attribute	Type of value contained	Number
Class of worker	Not in universe, Federal government, Local government, Private, ...	9
Education	Children, High school graduate, 9th grade, 10th grade, ...	17
Marital-status	Never-married, Married-civilian spouse present, Married-spouse absent, ...	7
Major industry code	Not in universe or children, Entertainment, Social services, Private household services, ...	24
Major occupation Code	Not in universe, Professional specialty, Other-service, Farming forestry and fishing, ...	15
Race	White, Black, Other, American Indian Aleut or Eskimo, ...	5
Sex	Female, Male	2
Detailed household and family stat	Child <18 never-married not in subfamily, Other-relative <18 never-married child of subfamily RP, Other-relative <18 never-married not in subfamily, ...	38
Detailed household summary in household	Child under 18 never-married, Spouse of householder, Non-relative of Householder, householder, Other-relative of householder, ...	8
Family members under 18	Both parents present, Neither parent present, Mother only present, ...	5
Country of birth father	Mexico, United-States, Puerto-Rico, Dominican-Republic, Jamaica, ...	42
Country of birth mother	India, Mexico, United-States, Puerto-Rico, Dominican-Republic, England, ...	42
Country of birth self	United-States, Mexico, Puerto-Rico, Peru, Canada, South Korea, India, Japan, ...	42
Citizenship	Native-Born in the United States, Foreign born-Not a citizen of US, ...	5

Table 8 | Similarities between attributes in adult

	Workclass	Education	Marital-status	Occupation	Relationship	Race	Sex	Native-country
Workclass	1.000	0.291	0.182	0.461	0.324	0.162	0.157	0.177
Education	0.291	1.000	0.335	0.428	0.287	0.128	0.2	0.045
Marital-status	0.182	0.335	1.000	0.194	0.654	0.161	0.232	0.288
Occupation	0.461	0.428	0.194	1.000	0.298	0.176	0.13	0.128
Relationship	0.324	0.287	0.654	0.298	1.000	0.182	0.236	0.248
Race	0.162	0.128	0.161	0.176	0.182	1.000	0.442	0.125
Sex	0.157	0.2	0.232	0.13	0.236	0.442	1.000	0.087
Native-country	0.177	0.045	0.288	0.128	0.248	0.125	0.087	1.000

and the relationship indicates a relationship with a family member. Since both of them are related to family relation, the similarity of them was high. The important point is these two attributes could be interpreted as semantically similar, so the result of them was not a big mistake.

4.3. Comparison between Similar Datasets of Census

4.3.1. Experiment

We calculated similarity between the Adult and the Census-Income that are considered to be the pair of similar datasets. The Census-Income has corresponding attributes for all seven string attributes in the Adult. In this experiment, the threshold value was set to 0.5, and pairs of attributes were determined as matched whose similarities exceeded it.

Table 9 | Similarities between attributes in adult and census-income

Attribute	Matching attributes	Similarity
Workclass	Class of worker	0.559
Education	Education	0.742
Marital-status	Marital-status	0.926
	Detailed household summary in household	0.648
	Detailed household and family stat	0.538
Occupation	Major occupation code	0.619
Relationship	Marital-status	0.709
	Detailed household summary in household	0.688
	Detailed household and family stat	0.637
	Major industry code	0.575
	Family members under 18	0.503
Race	Race	0.999
Sex	Sex	0.994
Native-country	Country of birth self	0.999
	Country of birth father	0.998
	Country of birth mother	0.998
	Citizenship	0.664

4.3.2. Result

Table 9 shows the comparison result with similarities that were 0.5 or more. In other combinations, similarities were about 0.2–0.5 or less. The result showed that they were almost correctly identified. The marital-status was similar to three attributes but the similarity with the marital-status was highest of them pre-eminently. The relationship had five results but it had no corresponding attributes in the Census-Income. The marital-status was similar for the same reason as the first experiment described in Subsection 4.2. The other attributes had some instances that contained semantically similar words such as child, married and family. These instances increased similarities with the relationship. The native-country had multiple results as similar attributes. However, since three of them were about someone's country of origin and it was reasonable that the similarity was high. Since the citizenship contains the word "United States", it made the similarity value high. Since similarities between other attributes had become lower, it could be considered that effective matching between datasets showing the same entity was possible.

4.4. Comparison between Similar Datasets of Human Resources

4.4.1. Experiment

We calculated similarity between the IBM HR and HR Data Set that are considered to be the pair of similar datasets. They both contain human resources data and they each have some attributes that may correspond each other. In this experiment, the threshold value was set to 0.5, and pairs of attributes were determined as matched whose similarities exceeded it.

Table 10 | Similarities between attributes in HR data Sets

Attribute	Matching attributes	Similarity
Gender	Sex	0.995
MaritalStatus	MaritalDesc	0.982
Job role	Position	0.663

Table 11 | Similarities between department and other attributes in HR

Attribute	Matching attributes	Similarity
Department	Department	0.467
	Position	0.423
	Employee source	0.293

4.4.2. Result

Table 10 shows the comparison result with similarities that were 0.5 or more. In other combinations, similarities were about 0.2–0.5 or less. The pairs of Gender and Sex, and MaritalStatus and MaritalDesc are regarded as matched with a high similarity score because they contain similar instances. However, the Department in both datasets could not be matched. Table 11 shows the top three comparison results between the Department in the IBM HR and all attributes in the HR Data Set. The similarity of the combination of Department was 0.467, which was lower than assumed. The difference of types of instances and the number of them caused low similarity despite the same attribute name. The pair of JobRole and Position was regarded as matched. The attribute names were similar but their meanings were different. But in fact, they both contained similar values like job type. Hence, this result was considered to be correct.

4.5. Comparison between Different Datasets

4.5.1. Experiment

We calculated similarity between the Adult and the Bank that are considered to be completely different datasets. However, the Bank has three attributes that are conceivable to mean the same attributes in Adult, which are job, marital and education in the Bank. Throughout this experiment, we investigated whether pairs of attributes representing the same entity could be found in different datasets. In this experiment, the threshold value was set to 0.5, and pairs of attributes were regarded as matched whose similarities exceeded it.

4.5.2. Result

Table 12 shows the comparison result with similarity is 0.5 or more. The result showed that job and marital were correctly identified. It seemed to be related to the fact that the types of the instances included in the attribute were similar in the Adult and the Bank. The attribute relationship in the Adult had no corresponding attribute in the Bank but it was similar to the marital for the same reason as the first experiment described in Subsection 4.2. Table 13 shows the top three comparison results between the education in the Adult and all attributes in the Bank. The similarity of the combination of education was 0.288, which was lower than assumed. Furthermore, the result of education was not the best. The difference of types of instances caused low similarity despite the same attribute name.

Table 12 | Similarities between attributes in adult and bank

Attribute	Matching attributes	Similarity
Marital-status	Marital	0.886
Occupation	Job	0.599
Relationship	Marital	0.610

Table 13 | Similarities between education in adult and attributes in bank

Attribute	Matching attributes	Similarity
Education	Month	0.363
	Job	0.298
	Education	0.288

5. DISCUSSION

From the first experimental result described in Subsection 4.2, the attributes that do not match but that are semantically similar gave the high similarity, such as the marital-status and the relationship. This result shows that complete automatically matching is difficult using Word2Vec. However, the marital-status and the relationship are semantically similar because they both indicate family information. This fact also indicates the possibility that using semantic similarity of instances is effective even when attribute names are completely different.

Our approach could find correct corresponding attributes in similar datasets that were the Adult and the Census-Income from the second experimental result described in Subsection 4.3. These two datasets contain the census data. However, the relationship was matched to five different attributes in the Census-Income. The marital-status is the marital information, the major industry code is about the major industry, and the other three are described about family information. Latter four attributes have instances containing common words with instances of the relationship. It was considered that the calculated vectors were close to each other. As a result, similarities with the relationship were high. However, the relationship in the Adult and four matched attributes except the major industry code in the Census-Income have family information so that they can be considered to be similar to one another.

We conducted one more experiment of comparison between similar datasets in Subsection 4.4. The datasets were the IBM HR and HR Data Sets, and they contain human resources data. The result of pair of Department showed that it is possible to find out the difference of attributes that were same attribute names but instances were significantly different. Furthermore, the result of JobRole and Position showed that it is possible to find the correspondence of attributes whose attribute names are different but instances are similar to each other. As a result, our approach could find correct correspondences considering semantics of instances, which were difficult to match using only attribute names.

The fourth experimental result described in Subsection 4.5 indicates that our approach had the possibility to find matching attributes between different datasets that were created for different purpose. However, the similarity about the combination of education was very low. This is because there is a big difference in the included values and the classification degree of information required for each datasets is different. The education in the Adult is very fragmented, while the education in the Bank is divided only into four categories. Furthermore, values of the education in the Bank are abstract such as “primary” and “secondary.” When calculating the education vector in the Bank, it is considered that these abstract words did not calculate a vector expression appropriate for the education. These differences of information required between the Adult and the Bank made so different vector and the similarity in the combination of education was so low.

6. CONCLUSION

In this paper, we proposed an instance-based schema matching approach using the Word2Vec as the semantic similarity. Our approach can apply to attributes that contain only string instances. We considered that values truly express semantic of their attributes, and we calculated the vector of attribute using values. We conducted four experiments; the first is the determination of the similarity threshold, the second and the third are the comparisons between similar datasets, and the fourth experiment is the comparison of different datasets. As a result, our approach showed the possibility of detecting the corresponding attributes by comparing vectors using Word2Vec. It showed possibility of detecting the correspondences between attributes that are semantically similar but not consistent, such as same attribute names but different instances or different attribute names but similar instances. However, it seems to be difficult to completely identify them automatically because the results have the cases that are wrong matching or 1–*n* matching not 1–1.

Future efforts will focus on improvement of accuracy. To devise a better method of calculating a vector of instances is needed, since we just add all the words included in an instance in this approach. The importance of words and the number of words in a value should be considered. Our approach can be applied to the attributes of character strings. Real datasets, however, contain attributes that categorized as keys such as one or a few letters expressed by numbers and alphabets. In this case, our approach cannot be applied at all because the Word2Vec is weak for words that are not in the corpus and those that have no semantics in itself. Thus, we need to accommodate to words that are not in corpus or consist of keys. In addition, our proposed method did not consider what words was contained in the used model. The used model might have the vectors of words that were composed of multiple words. However, such composed words could not be used because values were decomposed into single words in the calculation method. This problem prevented from more appropriate calculation of such values that contained composed words. If an instance has a composed word, it is necessary to use it as a composed word without decomposing. Furthermore, we need to solve the case that the matching result is 1–*n* matching. We introduced our method but did not compare with other methods. It is necessary to compare with other methods and verify the effectiveness of the proposed method quantitatively.

REFERENCES

[1] P.A. Bernstein, J. Madhavan, E. Rahm, Generic schema matching, ten years later, Proc. VLDB Endow. 4 (2011) 695–701.

[2] A.A. Alwan, A. Nordin, M. Alzeber, A.Z. Abualkishik, A survey of schema matching research using database schemas and instances, Int. J. Adv. Comput. Sci. Appl. 8 (2017) 102–111.

[3] T. Mikolov, W-t. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: Proceedings of NAACL-HLT, Association for Computational Linguistics, Atlanta, Georgia, 2013, pp. 746–751.

[4] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, pp. 1–12, arXiv:1301.3781v3.

[5] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems, volume 2, ACM Digital Library, Nevada, USA, 2013, pp. 3111–3119.

[6] K. Nozaki, T. Hochin, H. Nomiya, Semantic Schema Matching for String Attribute with Word Vectors, in: 6th International Conference on Computational Science/Intelligence and Applied Informatics, 2019, p. 6.

[7] H. Zhao, S. Ram, Clustering database objects for semantic integration of heterogeneous databases, in: AMCIS 2001 Proceedings, volume 70, 2001, pp. 357–362.

[8] J. Partyka, L. Khan, B. Thuraisingham, Semantic schema matching without shared instances, 2009 IEEE International Conference on Semantic Computing, IEEE, Berkeley, CA, USA, 2009, pp. 297–302.

[9] O.A. Mehdi, H. Ibrahim, L.S. Affendey, An approach for instance based schema matching with google similarity and regular expression, Int. Arab J. Inform. Technol. 14 (2017) 755–763.

[10] Google code archive - long-term storage for google code project hosting. https://code.google.com/archive/p/word2vec/ (accessed 2019-5-13).

[11] Uci machine learning repository. http://archive.ics.uci.edu/ml/datasets.html (accessed 2019-6-25).

[12] Kaggle. https://www.kaggle.com/ (accessed 2019-6-25).