Hindawi
Security and Communication Networks
Volume 2022, Article ID 2427155, 10 pages
https://doi.org/10.1155/2022/2427155

Research Article

Entity Relationship Modeling for IoT Data Fusion Driven by Dynamic Detecting Probe

Ye Tao,[1] **Shuaitong Guo**,[1] **Hui Li**,[1] **Ruichun Hou**,[2] **Xiangqian Ding**,[2] **and Dianhui Chu**[3]

[1]*College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, China*
[2]*College of Information Science and Engineering, Ocean University of China, Qingdao, China*
[3]*School of Computer Science and Technology, Harbin Institute of Technology (Weihai), Weihai, China*

Correspondence should be addressed to Ye Tao; ye.tao@qust.edu.cn

Received 13 March 2022; Accepted 6 May 2022; Published 8 July 2022

Academic Editor: Qi Li

To solve the problem of integrating and fusing scattered and heterogeneous data in the process of data space construction, we propose a novel entity association relationship modeling approach driven by dynamic detecting probes. By deploying acquisition units between the business logic layer and data access layer of different applications and dynamically collecting key information such as global data structure, related data, and access logs, the entity association model for enterprise data space is constructed from three levels: schema, instance, and log. At the schema association level, a multidimensional similarity discrimination algorithm combined with semantic analysis is used to achieve the rapid fusion of similar entities; at the instance association level, a combination of feature vector-based similarity analysis and deep learning is used to complete the association matching of different entities for structured data such as numeric and character data and unstructured data such as long text data; at the log association level, the association between different entities and attributes is established by analyzing the equivalence relationships in the data access logs. In addition, to address the uncertainty problem in the association construction process, a fuzzy logic-based inference model is applied to obtain the final entity association construction scheme.

1. Introduction

Data become an important resource in the information era. For different application scenarios, data can be stored in either centralized or distributed environment. It becomes particularly important to discover the association and correlation among heterogenous data source from multiple domains [1, 2]. After data are collected intensively, there are problems such as low sharing of original information, disconnection between information, and business processes and applications, which can easily lead to the formation of information silos [3]. In particular, the IoT industry requires huge technical data support to realize the procedural industrial processes and technologies, such as the construction of smart cities. This needs to solve the problem of information silos to achieve data sharing and fusion of centrally collected data under the premise of ensuring data security [4, 5]. To explore the correlation between data, some enterprises have started to build data space to integrate the data collected centrally, to eliminate information silos.

From the early days of data warehouses, data lakes, to the today's data fabric and data space systems, connecting data entities plays a vital role in data analysis. In the past, data association operations usually required cooperation between business-related personnel and database administrators to complete, which usually meant a lot of labor, material, and time, and its scalability was poor, and once the data changed, the data association information needed to be generated again. There is also a lot of research in academia on how to generate correlation information between data quickly and accurately. Current research results mainly focus on discovering associations between entities or attributes through the semantic matching of dictionaries or semantic libraries, using data representation, or content similarity judgments [6], and then using plain Bayesian learning algorithms to calculate the probability of similarity between data. Many of these methods have poor generalizability, slow response, and low accuracy when attempting to discover the existence of associations from a large amount of data.

In this paper, we propose a new approach to discover entity association relationships in big data. First, this approach obtains business logic information and database data through dynamic probes deployed between the business logic layer and data access layer of different systems. Then, it portrays the similarity degree among entities in three dimensions, schema, instance, and log and gives the similarity values among entities in these different dimensions. Finally, based on the fuzzy logic inference method [7, 8], the similarity values among entities in different dimensions are converted into normalized values that can be uniformly measured to obtain the best matching results of entity association. Thus, heterogeneous data from multiple source databases are integrated into a comprehensive enterprise data space through entity matching.

2. Related Work

In academic research, entity association is mainly divided into two types: schema matching [9] and instance analysis [10]. Schema matching extracts structural features from data sources as metadata and analyzes them to achieve association matching between data with fewer resources; matching based on instance analysis analyzes the data itself to obtain matching information, which usually consumes more resources but can obtain more accurate and comprehensive analysis results.

For schema matching academic research [11], Gomes dos Reis et al. used Structured Query Language (SQL) to extract features such as the name of the database, name of the schema, and type of column as metadata sets from each selected dataset. Then, they joined all metadata from each dataset into a metadata database. Finally, the correlation between the metadata was calculated by different methods to establish an association between the source data.

In addition to building a database through metadata [12], Berlin et al. use plain Bayesian learning to populate the attribute dictionary with example values provided by domain experts. To make efficient use of the attribute fields stored in the database, they employ statistical feature selection techniques to learn an efficient representation of the examples. With some columns of operations, the optimization process, which is based on a minimum cost maximum flow network algorithm, finds the overall optimal match between the two customer schemas based on the sum of the individual attribute matching scores.

For the ontology semantic similarity problem in schema matching, Meng et al. [13] studied the semantic similarity model based on distance, information content, and attributes and discovered that converting words to concept words in ontologies and performing semantic similarity calculations that can deliver the precise and effective measurement of targeted ontology semantics in a domain, which improves the accuracy of ontology semantic analysis in schema matching.

We can find that schema matching can effectively distinguish the association between data according to the analyzed information when processing a small amount of data, and the processing speed does not change significantly with the change of data volume because the analyzed elements are fixed, and the association matching between data can be achieved with less resources. However, when the amount of data grows exponentially, the probability that model information of different categories of data is similar or identical increases sharply because the amount of pattern information is certain, leading to a weakening of the differentiation effect of pattern matching analysis data.

In academic research on instance analysis, the preprocessing that mines associations between data include categorized data. For example, for the data conflict problem in data fusion, in [14], the conflict can be divided into two categories: uncertain conflict and contradictory conflict, and then the duplicate data of the same representation are fused, thus solving problems such as the possible conflict between different values for the same attribute. Reference [15] proposes that solutions such as name and description matching in schema matching can be used for element-level instance analysis. Addressing the problem of different data types in instances, the authors in [16, 17] propose an approach for classifying instance data and present a systematic theoretical framework for establishing data associations for different classes of data. Instance analysis can maintain a better differentiation of data fusion when dealing with large amounts of data, but this often takes a long analysis time. In addition, instance analysis often takes a lot of time and operational resources to correct data association relationships when data change, especially when new data are added.

Academics are also studying the integration of deep learning with logs, for example, using deep learning to replace statistical methods in logs that portray associations between users and certain types of items or certain things. Mohanty et al. [18] cleaned the web log files collected by the IoT, built user profiles, saved similar information, and proposed a recommendation system based on rough fuzzy clustering to recommend e-commerce shopping sites to users. The logs contain correlations among the data, but they are generated by manipulating the data, and only part of the data are involved compared to the overall data, leading to a lack of completeness, and their analysis is unable to explore the correlations that exist in all the data. In this paper, we offer a proposal for extracting the data association information in logs and using it as a basis for matching entity associations in multisource data to construct entity associations in enterprise data space; this approach builds on the feature that logs contain association information between data [19].

The constantly increasing amount of data accumulating in the development of enterprises leads to an increasing size and number of categories of data, and methods such as schema matching, instance analysis, and log mining to analyze data from a single dimension may have problems such as not making full use of the diversity of data or incomplete analysis. Addressing the above issues, this paper analyzes the data from multiple dimensions by integrating schemas, instances, and logs to make full use of the diversity of data to establish entity associations.

3. Our Customized Framework

The entity association model in Figure 1 shows the mapping relationship between multiple sources of data from different departments in the enterprise business system and the data space. According to the multidimensional analysis framework proposed in this paper, normalized similarity values between data that can be compared are obtained to establish the association relationship between entities. As shown in Figure 1, R_1 indicates a similarity value of 1 between its associated entities a_{13} and n_{11}.

In the middle of the business logic layer and data access layer of each business system, such as enterprise resource planning (ERP), customer relationship management (CRM), and software configuration management (SCM), we deploy probes to obtain data. Then, the business logic layer of the data is stored as logs, and the rest of the data are stored in a relational database. To overcome the problems of large size and a variety of data types, the model preclassifies the data based on their characteristics and nature, which improves the data processing and increases the accuracy of matching between entities. The structure and content of the data are divided into two categories: schema and instance, while logs as a carrier of business logic are grouped into a separate category. The similarity values between the data are analyzed and calculated in three dimensions: schema, instance, and log. The schema matching analysis includes both attribute names and constraints, and the instance analysis is divided into three analysis methods according to data type: numeric, character, and long text. Based on the attribute association information contained in SQL, the log analysis calculates the similarity values among the data. Finally, based on the fuzzy logic analyzer, a normalization calculation is performed based on similar values for the data in different dimensions to obtain the effective association values in the data space. The corresponding schema is shown in Figure 2.

3.1. Schema Similarity Model. Many different databases are developed by database designers to fit application scenarios, naming conventions, and other factors, but database designs generally contain table and field names, table structures, and data types. As such, the attribute names and constraints of the schema information in the database are extracted as the analysis content of the schema similarity model to measure the similarity between the data.

3.1.1. Name of Attribute. Attribute name analysis is divided into two types: plain text similarity and text semantic similarity analysis. The text similarity between attribute names is calculated by the edit distance algorithm, and text semantic similarity is calculated through a semantic library.

Edit distance is a way of quantifying how similar two strings are; it takes two words w_1 and w_2 and finds the minimum number of operations required to convert w_1 to w_2. The plain text similarity value is defined according to the minimum number of edits, as shown in the following equation:

$$S_{\text{plain}}(w_1, w_2) = 1 - \frac{D(w_1, w_2)}{\text{Max}(l_1, l_2)}, \tag{1}$$

where l_1 and l_2 are the character lengths of w_1 and w_2 and D is the edit distance of w_1 and w_2.

Different expressions may be used for the description of the same entity. For example, if the information of an upstream company is recorded in the enterprise database, its attribute name can be named CompanyID and SupplierID based on different scenarios. To address the fact that plain text analysis cannot resolve the semantics between words, a semantic-based similarity analysis method is proposed. In particular, a tree semantic hierarchy is established for the attribute names, as shown in Figure 3, and the similarity between words is calculated by the corresponding positions of the attribute names in the tree diagram.

Therefore, the equation calculating the semantic-based similarity is

$$S_{\text{sema}}(w_1, w_2) = \frac{2H}{N_1 + N_2 + 2H}, \tag{2}$$

where N_1 and N_2 denote the shortest paths from words w_1 and w_2 to the nearest common parent word w, respectively, and H denotes the shortest path from w to the root node.

S_{name} is defined as the maximum of the plain text similarity and the semantic similarity of the text, as shown in the following equation:

$$S_{\text{name}} = \text{Max}\left(S_{\text{plain}}, S_{\text{sema}}\right). \tag{3}$$

3.1.2. Constraint. Designers follow certain principles when programming columns in a database, such as the appropriate data type and whether it is empty. The representative constraints selected from these rules can be used to explore the similarity among columns. Constraints listed in Table 1 are extracted as features: type of each column, if the column is a primary or foreign key or not if the column has constraint of null or not null if the column has comments.

In the following equation, we assume that the two columns requiring constraint similarity discrimination are A and B, and a_i and b_i are the values of the ith candidate constraint corresponding to the attributes of the two columns, respectively, such that

$$v_i = \begin{cases} 1 & a_i = b_i \\ 0 & \text{otherwise} \end{cases}, \qquad i = 1, 2, \ldots, n, \tag{4}$$

where n is the number of candidate constraints. Therefore, the attribute constraint similarity between column A and column B is calculated by

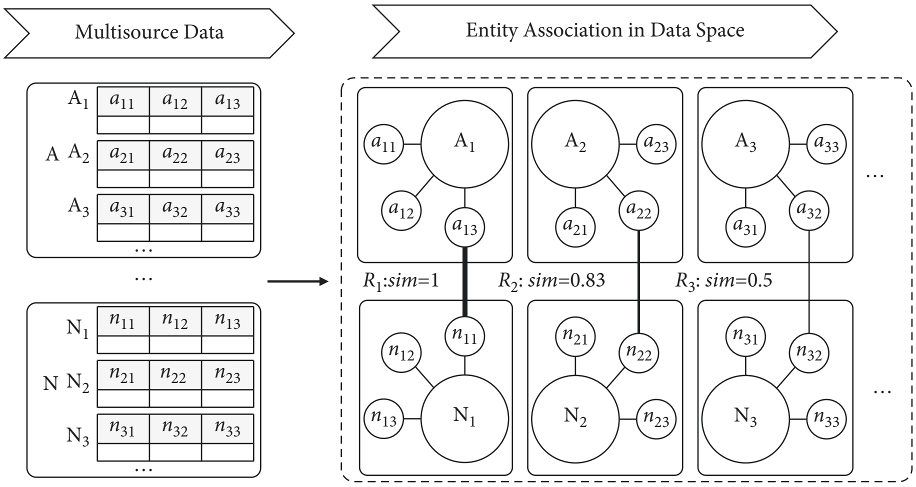

Multisource Data
Entity Association in Data Space
A
A_1 a_{11} a_{12} a_{13}
A_2 a_{21} a_{22} a_{23}
A_3 a_{31} a_{32} a_{33}
N
N_1 n_{11} n_{12} n_{13}
N_2 n_{21} n_{22} n_{23}
N_3 n_{31} n_{32} n_{33}
R_1:sim=1
R_2: sim=0.83
R_3: sim=0.5
…

Figure 1: Entity association mapping for multisource data.

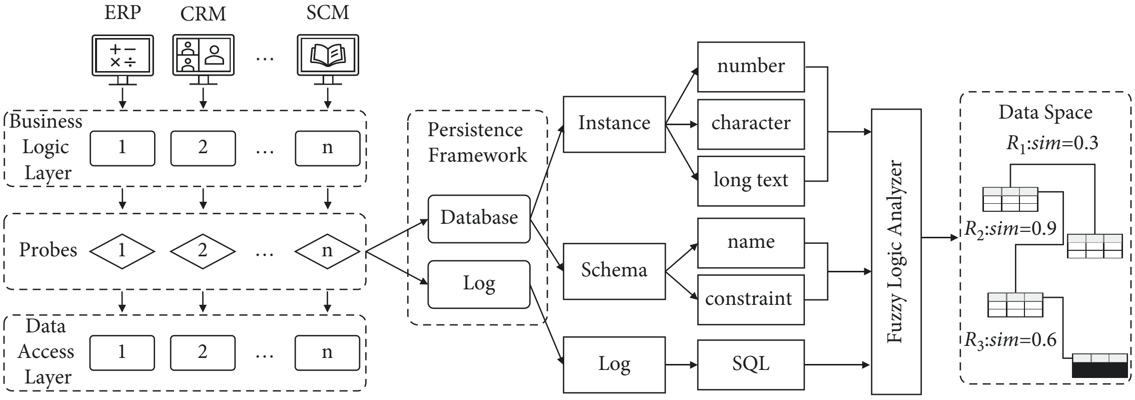


Figure 2: A logical framework for multisource data analysis.

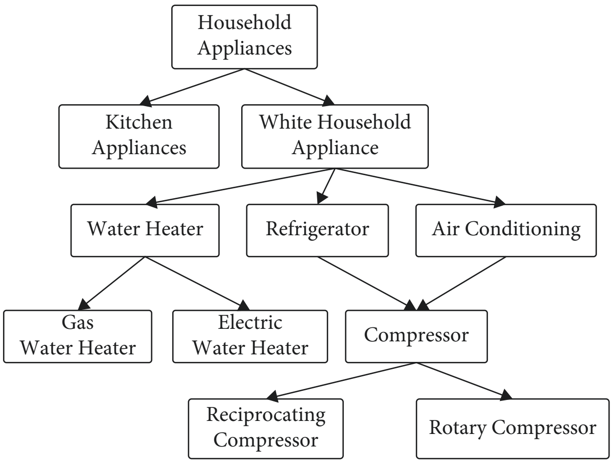


Figure 3: Attribute name tree semantic hierarchy diagram.

Table 1: Constraint features.

$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
Type of column	Null	Primary key	Foreign key	Comments

$$S_{\text{cons}} = \frac{\sum_i v_i}{n}. \tag{5}$$

3.1.3. Schema Similarity. S_{schema} includes attribute names and constraint analysis of similar values by weighting, as shown in the following equation:

$$\underset{\text{schema}}{S} = \alpha S_{\text{name}} + (1-\alpha) S_{\text{cons}} \, (\alpha \in [0,1]). \tag{6}$$

3.2. Instance Similarity Model. Since there are similarity trends in datasets representing similar entities, such as value intervals, extreme values, and keywords. Similarity relationships between data can be established by the main content of the dataset. It is obvious that data categories have distinctive features of a dataset, and differences in data categories lead to variability in the attributes chosen to characterize the dataset. Establishing differentiated feature extraction schemes for different classes of datasets can improve the accuracy of data association matching. The data types in the database are categorized, and different categories of data correspond to different processing schemes; generally, if the data categories are different, there is no similar relationship.

According to the different data types, instance analysis can be divided into the following three types: numeric, character, and long text. The numeric type refers to the exact numeric data type and the approximate numeric data types in Table 2. The string data types are divided into two categories, character, and long text, according to the length of the text. After classifying and clustering the data, the similarities between the data are analyzed according to the process shown in Figure 4.

3.2.1. Number. For scalar data, the similarity between columns can be evaluated from the perspective of numerical distribution, e.g. the median, mean, variance and etc. In order to reflect the characteristics of numerical scalars from different aspects, the selection of features is focused on the following three aspects, the maximum and minimum values that can define the range of data, the mean, arithmetic median and plural that reflect the main distribution of data, the sample standard deviation that can reflect the degree of dispersion of data, these indicator elements are not sensitive to the change of data volume and can be used as the feature elements for calculating column similarity, while the number of non-null values and the cumulative sum of the data do not change significantly with the change of data volume, but are not suitable as feature elements. Finally, the feature vector corresponding to each column is calculated, substituted into the cosine similarity formula, and the result is used as the numerical similarity value.

3.2.2. Character. Character is short textual content, and it uses the term frequency-inverse document frequency as the similarity calculation algorithm. First, the content of the columns that need to determine similarity is combined as a separate dataset. Then, the vectors for each column are found. Finally, the feature vectors are substituted into the cosine similarity formula to calculate the similarity value.

3.2.3. Long Text. Long text is long text content, where the records in the columns are mapped as vectors, a model is built using an autoencoder, and the similarity values among columns are calculated based on the model. Assuming that A and B are the two columns in the database, and they share the long text data type (Figure 5). The overfitting problem of the model due to the large difference in the number of datasets is solved by randomly selecting k records in columns A and B as the sample data sets S_1 and S_2. Since vectors are required as input for the autoencoder, the text in the sample data sets is transformed into vectors $\overrightarrow{U}$ and $\overrightarrow{V}$. Then, the vectors are divided into a training set and test set, the autoencoder model is built using the training set, and the similarity of columns A and B is calculated according to the accuracy of the test set.

Table 2: Data type categorization.

Data type	Members
Exact numeric data type	Smallint, mediumint, int, bigint
Approximate numeric data type	Float, double, decimal
String data types	Char, varchar, blob, text

The autoencoder model calculates similarity, as shown in Algorithm 1. For input, x is divided into a training set and a test set according to a custom scale, y is used as the test set, and ω is the custom text similarity threshold. On output, λ and θ are the percentages of the test set evaluated as similar. For autoencoder 1, x and y for the input in Algorithm 1 are $\overrightarrow{U}$ in vector space and the test dataset of $\overrightarrow{V}$ in vector space, and the output is λ_1 and θ_1. For autoencoder 2, x and y for the input in Algorithm 1 are $\overrightarrow{V}$ in vector space and the test dataset of $\overrightarrow{U}$ in vector space, and the output is λ_2 and θ_2. According to the results obtained from the autoencoder, S_{long} represents two columns of similar values, as shown in the following equation:

$$S_{\text{long}} = \text{Min}\left(\frac{\theta_1}{\lambda_1}, \frac{\theta_2}{\lambda_2}, 1\right). \tag{7}$$

3.3. Log Similarity Model. The business logic layer in the layered architecture mainly packages the attributes and behaviors of entities. Although the representation of entities varies across different business logics and similar entities have similar attributes and behaviors. The SQL commands recorded in the logs contain correlation relationships among columns, which can be used as a basis of analysis for measuring column similarity. The column-to-column similarity can be obtained by counting the number of equivalence relations in the log file.

In the following equation, we assume that A and B are columns in the database, and the log similarity value of columns A and B is calculated by

$$S_{\log} = \frac{N_{ab}}{N_a + N_b}, \tag{8}$$

where a and b are the names of columns A and B, N_a and N_b are the number of SQL commands containing a and b in the log, and N_{ab} is the number of SQL commands containing both a and b in the log.

3.4. Fuzzy Logic Similarity. According to the previous section, the calculation of the data with the proposed model can obtain similar values in three dimensions: pattern, instance, and log, which need to be unified into directly comparable values since similar values on different dimensions are not directly comparable. The methods that can generally be used to convert multidimensional values into a single value are the Delphi method [20], weighted average, and fuzzy logic. Delphi method relies on domain-specific knowledge, and when the data source is not regular, it cannot be well adapted to the data, while weighted average, due to its fixed form, is

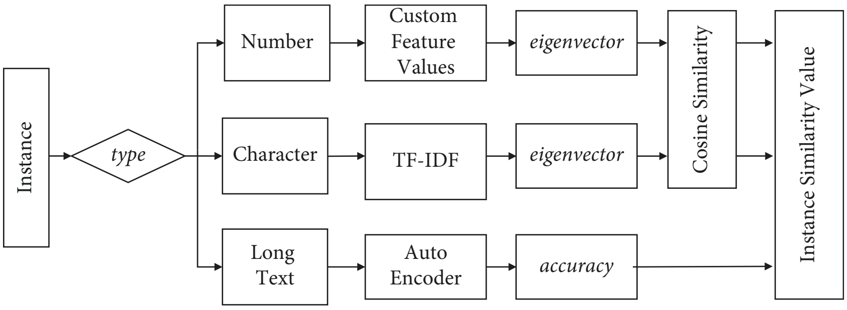


Figure 4: Instance analyzer.

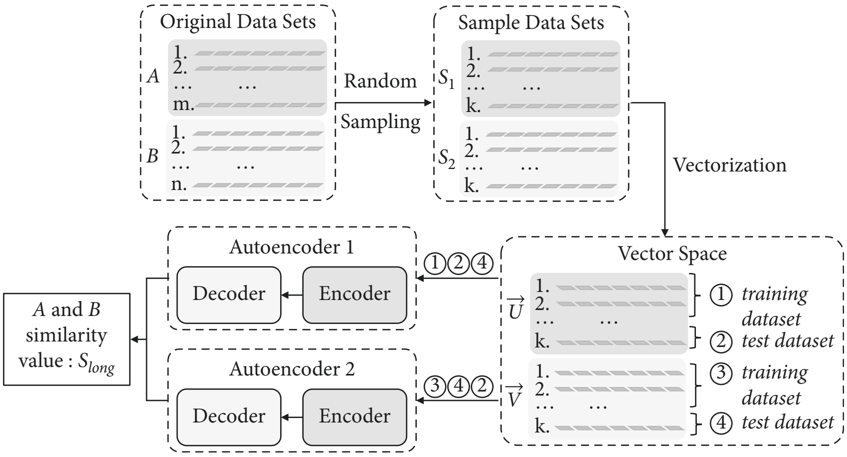


Figure 5: The long text analysis process.

```
Input : $x, y, \omega$
Output : $\lambda, \theta$
$a$_train, $a$_test←train_test_split $(x)$
$b$_test←$y$
$a$_num, $b$_num←len $(a$_test$)$, len $(b$_test$)$
input←$a$_train
encoded = Dense (input)
decoded = Dense (encoded)
autoencoded = Model (input, decoded)
$a$_test_predict = autoencoded $(a$_test$)$
$b$_test_predict = autoencoded $(b$_test$)$
    For$a$, $b$in$a$_test, $a$_test_predict
        $s$_$a$_num + +←similarity $(a, b) \geq \omega$
    $\lambda$←num/$s$_$a$_num
    For$a$, $b$in$b$_test, $b$_test_predict
        $s$_$b$_num + +←similarity $(a, b) \geq \omega$
    $\theta$←num/$s$_$b$_num
```

Algorithm 1: Long text similarity calculation method.

more homogeneous for data processing and cannot make full use of the characteristics of the data. In contrast, fuzzy logic can contain expert domain knowledge [21] and its ability to use multiple functions for data fitting when processing data. Its adaptability is relatively good, so fuzzy logic is chosen to normalize the similar values of multiple dimensions.

The similarity values obtained from the above calculation by schema, instance, and log similarity models are processed using fuzzy logic for standardization. While A and B are the columns in the database, the similarity values obtained from the above three-dimensional analysis are substituted into the affiliation function to obtain the affiliation values. The values that meet the fuzzy rules are aggregated according to the rules and defuzzified to obtain a normalized measure of column-to-column similarity. In Figure 6, for example, A and B have similar values of 0.6, 0.7, and 0.8 in the schema, instance, and log dimensions. Through a series of fuzzy operations, the similarity value between A and B is 0.71.

4. Experiment

To verify the feasibility of the proposed framework, this paper uses data from all business systems of a company and stores them in a unified manner. The hardware environment for the experiments is an Intel(R) Xeon(R) Silver 4210 CPU @ 2.20 GHz, 64 GB RAM, and RTX2080Ti*4. The results are the average of three replicated experiments. The dataset consists of Haier, the upstream and downstream of Haier's supply chain, and public data set available on the Internet [22]. It mainly includes the following categories: product data, enterprise operation data, value chain data, and external data.

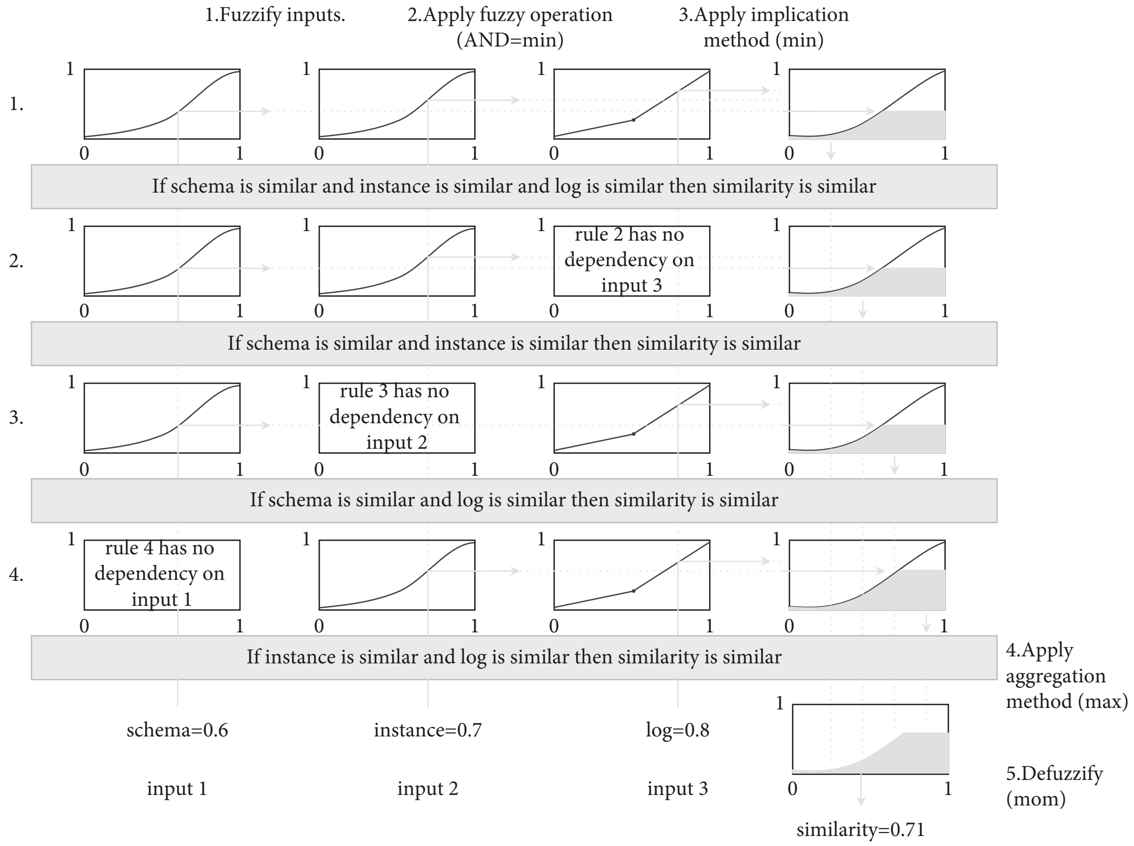

1.Fuzzify inputs.
2.Apply fuzzy operation (AND=min)
3.Apply implication method (min)
1.
If schema is similar and instance is similar and log is similar then similarity is similar
2.
rule 2 has no dependency on input 3
If schema is similar and instance is similar then similarity is similar
3.
rule 3 has no dependency on input 2
If schema is similar and log is similar then similarity is similar
4.
rule 4 has no dependency on input 1
If instance is similar and log is similar then similarity is similar
4.Apply aggregation method (max)
schema=0.6
instance=0.7
log=0.8
input 1
input 2
input 3
5.Defuzzify (mom)
similarity=0.71
0
1

Figure 6: Fuzzy logic instance diagram.

4.1. Data Aggregation Matching Experiments. Data aggregation matching experiment is as follows: the data provided by the suppliers are analyzed for correlations between the data using the model designed in this section, and the data are automatically imported into the summary table (the attributes to be imported and their corresponding partial data need to exist in the summary table in advance). Table 3 shows the matching results of 3000 records in the selected supplier data, respectively, where each row is the matching result information of each supplier. The total number of attributes refers to the total number of data attributes provided by suppliers, the total number of valid attributes refers to the data that can correspond to a column attribute in the summary file, and the total number of correctly associated attributes refers to the number of columns that are correctly integrated into the summary file after matching each supplier's data through the model. The correct rate is the ratio of the number of correctly associated attributes to the total number of valid attributes.

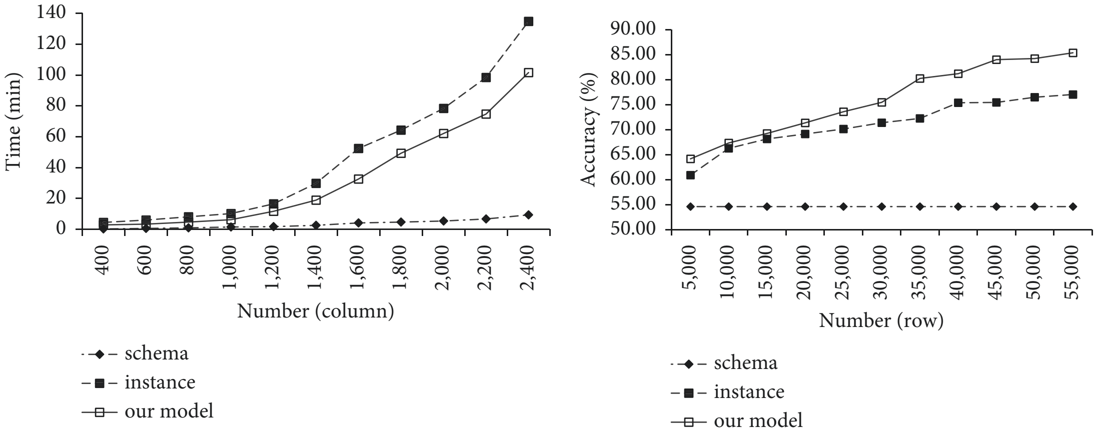
From the results, we can see that the best performance of data matching accuracy can reach about 89%, which can be well used as an auxiliary tool for data matching, while the analysis of the results of the lower accuracy of data matching for no. 2 reveals that more proprietary names and abbreviations are used in the data provided by its suppliers, and because its business involves relatively single, the content similarity is high, which leads to the low accuracy of pattern matching results in model analysis, thus leading to unsatisfactory results, and the accuracy can be subsequently improved by optimizing the semantic analysis in pattern matching.

4.2. Comparison of Experiments for the Different Solutions of Data Space Entity Association. A certain number of columns are randomly selected as samples from all data, and experiments are conducted using schema matching (see Section 3.1), instance analysis (see Section 3.2), and the fuzzy logic-based model proposed in this paper to compare them in two ways: running time and accuracy.

The Design of Experiment. (1) Runtime with different methods: from a total of 3702 columns, randomly select 400, 600, . . ., 2400 columns, 11 groups in total, record the running time of each set of data under the three methods, and repeat the above operation three times.Figure 7(a) shows the average runtime with different methods. (2) Accuracy with different methods: a total of 3702 columns are grouped according to numbers, characters, and long text; and 50 pairs of related columns are randomly selected from each group. 5000, 10000, . . ., 55000 rows were selected from the selected columns; 11 groups in total; and the similarity value of each group of data under the above three models was calculated to determine whether the prediction was correct according to the threshold value w. The percentage of

Table 3: Data matching results.

No.	Number of attributes	Number of valid attributes	Number of valid attributes correctly associated	Accuracy rate (%)
1	1847	942	749	79.51
2	2084	642	411	64.02
3	1974	762	647	84.91
4	1639	849	758	89.40



(a) (b)

Figure 7: Comparison of experiments for different methods: (a) running time of different methods and (b) accuracy of different methods.

correct prediction is calculated, and Figure 7(b) shows the average prediction accuracy of the above three methods.

Analysis of the Experiment. Experiments based on the schema take less time, as shown in Figure 7(a). The instance-based method takes significantly more time for the same amount of data due to the comprehensive content analysis, while the method proposed in this paper includes instance analysis but takes less time than the instance-based method because the data are analyzed in categories during the instance analysis.

The accuracy of the schema-based method was highest when the experimental sample was below 600, as seen in Figure 7(b). The method proposed in this paper maintained the highest accuracy after 800 columns, the schema-based method was limited after the data volume was 1400 columns due to the limited analysis elements, and the data matching accuracy decreased due to the increase of homogeneous data caused by easily mismatching events when the data size became larger. Overall, with the increase of data volume, the data matching accuracy of all analysis methods tends to increase, which is due to the fact that, in equal proportion sampling, when the sample is small, the number of similar data corresponding to the suppliers is smaller, which leads to the possibility of mismatching; and when the proportion of sampled data covering the overall data increases, the mismatching situation decreases significantly, and thus the correct rate of data matching gradually increases.

As shown in Figure 7, the proposed method in this paper can obtain a high accuracy rate in a short time with a moderate amount of data.

4.3. Long Text Validation Experiments. To study the performance of the autoencoder on the long text case in the instance analysis, two columns of associated long text are selected, and the performance of the model proposed in this paper is observed in different cases by changing the vector dimension.

The Design of Experiment. From the existing long text columns, 10 pairs with suitable amount of data and correlation were selected, and the running time and prediction accuracy under long text analysis were recorded by changing their vectorized coding length to 128, 256, 512, and 1024, and the results are shown in Figure 8.

Analysis of the Experiment. The higher the dimensionality is, the more time the experiment takes for the same amount of data as shown in Figure 8(a). Figure 8(b) shows that in the case of a small volume of data, if the dimensionality is too high, it will reduce the accuracy. The reason for this performance can be found by analyzing the principle of the autoencoder. The autoencoder model reduces the dimension of data to extract key information, and when the data size is small, the compressed extracted data features in the long text are limited, so the high-dimensional feature vector will be mixed with a large amount of noisy data, which results in a low accuracy rate. As the volume of data increases, more data features can be extracted from long text, and the high-dimensional feature vector can represent the text better and therefore obtain higher accuracy.

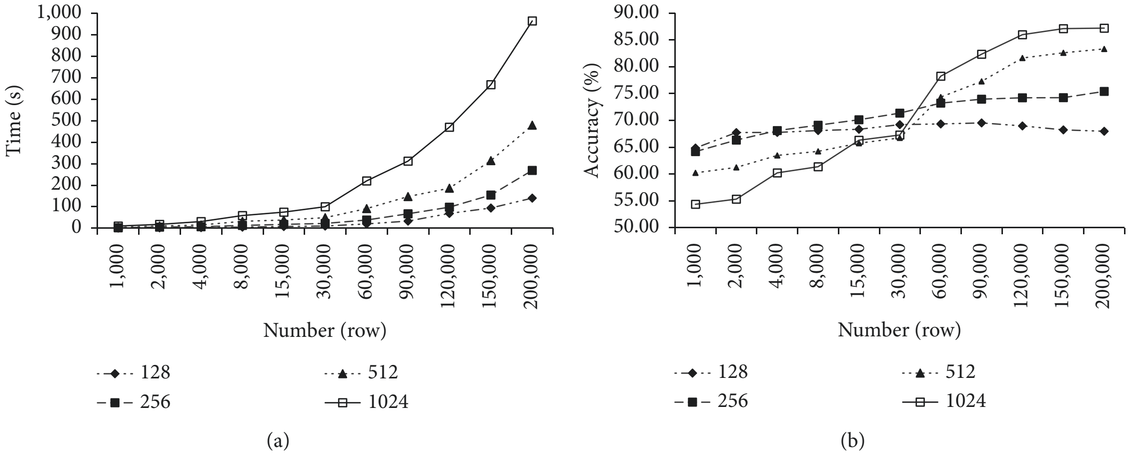


Figure 8: Comparisons of experiments for long text columns: (a) running time of different vector dimensions; (b) accuracy of different vector dimensions.

5. Conclusion

This paper proposes a hybrid data matching model based on schema, instances, and logs. The model consists of four main components: the front probe to acquire the analysis data, the analysis data, the three-dimensional outputs, and the normalized metric based on fuzzy logic. Experimental results show that the model provided in this paper has better results in terms of accuracy and efficient handling of mass data compared to previous single matching methods based on schema or instances. For further research, the focus is on how to establish a mapping relationship between data and weights and on establishing a guidance scheme for weight assignment to better address the impact of the randomness of multisource heterogeneous data on the accuracy of the results.

Data Availability

The data are generated by the actual operation process of the enterprise and involve private data that cannot be desensitized and can only be used internally for the time being.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1702902 and in part by the Shandong Province Colleges and Universities Young Talents Initiation Program under Grant 2019KJN047.

References

[1] Y. Tian, T. Li, J. Xiong, M. Z. A. Bhuiyan, J. Ma, and C. Peng, “A blockchain-based machine learning framework for edge services in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1918–1929, 2022.

[2] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, “Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles,” *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[3] J. Lv, K. Shen, S. Johnson, F. Chen, and G. Li, “Application on information island with information visualization and software engineering,” in *Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI)*, pp. 598–603, Nanjing, China, November 2018.

[4] Y. Tian, Z. Zhang, J. Xiong, L. Chen, J. Ma, and C. Peng, “Achieving graph clustering privacy preservation based on structure entropy in social IoT,” *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2761–2777, 2022.

[5] J. Xiong, R. Ma, L. Chen et al., “A personalized privacy protection framework for mobile crowdsensing in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[6] M. Nordin, A. Alzeber, and A. Zaid, “A survey of schema matching research using database schemas and instances,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 10, 2017.

[7] X. Li, H. Wen, Y. Hu, and L. Jiang, “A novel beta parameter based fuzzy-logic controller for photovoltaic MPPT application,” *Renewable Energy*, vol. 130, pp. 416–427, 2019.

[8] Z. Roumila, D. Rekioua, and T. Rekioua, “Energy management based fuzzy logic controller of hybrid system wind/photovoltaic/diesel with storage battery,” *International Journal of Hydrogen Energy*, vol. 42, no. 30, pp. 19525–19535, 2017.

[9] W. Tan and A. Mapforce, *Approximation Algorithms for Schema-Mapping Discovery*, vol. 42, 2017.

[10] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, “Multi-instance learning with discriminative bag mapping,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1065–1080, 2018.

[11] D. Gomes dos Reis, M. Ladeira, M. Holanda, and M. de Carvalho Victorino, “Large database schema matching using data mining techniques,” in *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW) 2018*, pp. 523–530, Singapore, November 2019.

[12] J. Berlin and A. Motro, “Database schema matching using machine learning with feature selection,” *Notes on Numerical*

Fluid Mechanics and Multidisciplinary Design, pp. 452–466, Springer, Berlin, Germany, 2002.

[13] L. Meng, R. Huang, and J. Gu, "A review of semantic similarity measures in WordNet," *Int. J. Hybrid Inf. Technol.*vol. 6, pp. 1–12, 2013.

[14] A. Bakhtouchi, "Data reconciliation and fusion methods: a survey," *Applied Computing and Informatics*, vol. 18, no. 3/4, pp. 182–194, 2022.

[15] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *The VLDB Journal*, vol. 10, no. 4, pp. 334–350, 2001.

[16] X. Xu and W. Wang, "Attribute identification between spatial datasets based on instance statistical similarities," in *Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing 2008*, pp. 1–5, Dalian, China, October 2008.

[17] E. Sutanta, R. Wardoyo, K. Mustofa, and E. Winarko, "A hybrid model schema matching using constraint-based and instance-based," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 3, pp. 1048–1058, 2016.

[18] S. N. Mohanty, J. Rejina Parvin, K. Vinoth Kumar, K. C. Ramya, S. Sheeba Rani, and S. K. Lakshmanaprabu, "Optimal rough fuzzy clustering for user profile ontology based web page recommendation analysis," *Journal of Intelligent and Fuzzy Systems*, vol. 37, no. 1, pp. 205–216, 2019.

[19] Y. Tao, S. Guo, C. Shi, and D. Chu, "User behavior analysis by cross-domain log data fusion," *IEEE Access*, vol. 8, pp. 400–406, 2020.

[20] I. Belton, A. MacDonald, G. Wright, and I. Hamlin, "Improving the practical application of the Delphi method in group-based judgment: a six-step prescription for a well-founded and defensible process," *Technological Forecasting and Social Change*, vol. 147, pp. 72–82, 2019.

[21] C.-L. Wu, T.-W. Ke, and T.-H. Meen, "Evaluation of intensified colorectal cancer treatment using model based on Delphi method, fuzzy logic, and analytical hierarchy process (DFAHP)," *Sensors and Materials*, vol. 33, no. 10, pp. 3499–3512, 2021.

[22] Industrial-Datasets, "Haier's internal dataset and publicly available datasets on the Internet," 2021, https://github.com/forgstfree/Industrial-Datasets.

[23] Y. Tao, S. Guo, R. Hou, X. Ding, and D. Chu, "Entity relationship modeling for enterprise data space construction driven by a dynamic detecting probe," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, J. Xiong, S. Wu, C. Peng, and Y. Tian, Eds., pp. 185–196, Springer International Publishing, New York, NY, USA, 2021.