JUORS

JOIN and UNION operations recommendation system

Chen Guan You
Department of Finance
National Taiwan University
Taipei, Taiwan
b08703112@ntu.edu.tw

Wen Pei Te
Department of Agronomy
National Taiwan University
Taipei, Taiwan
r10621204@ntu.edu.tw

He Shan Syue
Department of Agronomy
National Taiwan University
Taipei, Taiwan
r10621203@ntu.edu.tw

ABSTRACT

Schema matching finds similar columns from distinct sources, and it is widely used in data cleaning and integration. Generally, tables can be combined when they shared the same information. In database management system, JOIN and UNION are operations for combining tables. In this study, we developed JOURS, a JOIN and UNION operations recommendation system. With JOURS, similar columns from two tables can be identified, and the similarity score of two columns is calculated. JOURS supports both string and numeric data type columns and compares columns based on their column name. When users import two tables, a recommended table would be generated by JOURS. The recommended table contains the details of columns and the corresponding similarity scores, and it can be a reference when combining tables.

KEYWORDS

Schema matching, Entity matching, Data integration, instance-based method, K-Means clustering, JOIN operation, UNION operation

1 INTRODUCTION

In database management system, JOIN and UNION are operators for combining tables. In the early stage of database development, the database architecture designers planned and implemented database management systems. Therefore, they knew how to use JOIN or UNION operations properly and reasonably. However, the rising needs for database integrations make JOIN or UNION operations more difficult, due to the following reasons. Firstly, the integration of huge and complicated DBMS is hard to be able to fully understand the whole structure, not to mention further operations. Also, it is time-consuming to figure out all the relations and structure.

To determine whether JOIN or UNION operations can be used, schema matching can be adopted. Schema matching is an important operation in data cleaning and integration, and it aims to find columns with the same meaning from distinct sources of tables. There are two levels of matching, including schema-level and instance-level matching. Schema-level matching creates the attribute pair by comparing the meaning of the attribute name or the domain of the attribute. On the other hand, instance-level matching finds the overlaps of values or the same distribution for the quantitative data and identifies similar words based on the semantic or syntactic for the qualitative data. The similarity score is used to quantify the similarity between two columns.

Many studies have been worked on schema matching. MF-Join supports fuzzy word similarity join efficiently, and it considers two levels of matching, including element-level matching and record-level matching, and it supports fuzzy word similarity join efficiently [1]. Word2Vec is a word embedding methodology, and it is an algorithm that converts strings to vectors [2]. Word2Vec can be used to conduct the semantic comparison.

In this study, we implement JOURS, a JOIN and UNION operations recommendation system. We want to help database users to understand the relationship among attributes of different tables. And we achieve this goal by implementing an automatic matching system. When there are two tables that we don't know their relationship, this system will first recognize the data type of each column and then use a specific method and algorithm to match different attributes. Once the system finished all the analysis, it will create a recommendation table, and database users can view this table to understand which attributes are related to each other. After database integration is determined, one can apply JOIN or UNION operations more conveniently by the recommendation table. We believe that this system will save users a lot of time from checking relationships manually. And more details of implementation will be in the **METHOD** Part.

2 METHOD

JOURS is designed for data integration, and users need to import two tables in JOURS. In JOURS, one is called Source Table, and the other is Test Table. To find similar columns in Source Table and Test Table, the pipeline goes as follows:

2.1 PIPELINE

1. Load two tables (Source Table and Test Table) into JUORS. The source of tables can be local (as CSV files) or in MySQL.
2. Extract the feature vector for each column in Source Table and Test Table, and the detailed description is in **FEATURE EXTRACTION**.

3. Use the K-means Clustering algorithm to cluster columns from Source Table according to their feature vectors. The number of clusters can be determined by users. The default number of clusters in JOURS is 2 for both string and numeric type columns.
4. Add columns from Test Table into the most similar cluster.
5. Use schema matching techniques to match the columns of the same clusters, and we can match columns by their values or their column names. The detailed descriptions are in **NUMERIC MATCHING**, **STRING MATCHING**, and **COLUMN NAMES MATCHING**.
6. Recommend Table is created and Recommend Table records the information of columns and the corresponding similarity score. Recommend Table can be further exported as a CSV file or in MySQL.

Following are the detailed descriptions.

2.2 FEATURE EXTRACTION

To cluster the columns, we converted the columns to feature vectors. As far as we are concerned, the similar structure of columns may contain similar information. For the string type columns, the feature vector is [numeric ratio, character ratio, space symbol ratio, bracket symbol ratio, hyphen symbol ratio, slash symbol ratio, dot symbol ratio, comma symbol ratio]. For the numeric type columns, the feature vector is [mean, variance, coefficient of variation, minimum, maximum, Q1, median, Q3, range]. After converting to feature vectors, K-Means Clustering can then be applied.

2.3 NUMERIC MATCHING

For numeric matching, we used feature vectors that were generated from previous steps to calculate cosine similarity, and we use cosine similarity as the matching basis. Cosine similarity considers the cosine of angle between two vectors [3], and cosine similarity is calculated as

$$\cos(x, y) = \frac{x \cdot y}{\|x\|\|y\|} \tag{1}$$

where x and y are the feature vectors of two columns, $\|x\|$ is the Euclidean distance of vector x, and $\|y\|$ is the Euclidean distance of vector y. The value of cosine similarity is [-1, 1].

In order to get the same scale as other measures, we rescale the similarity from [-1, 1] to [0, 1]. However, there is one thing that feature-based similarity ignores, which is whether the two columns have the same distribution? Because even though the two columns have the same mean and variance, their distributions might be totally different. Therefore, we further use the KS test to test whether the two columns' distribution is the same or not, and the similarity score of numeric columns is calculated as

$$numeric\ score = \begin{cases} \cos(x, y) \times 1, if\ P - Value\ is\ [0.05, 1] \\ \cos(x, y) \times \frac{2}{3}, if\ P - Value\ is\ [0.01, 0.05) \\ \cos(x, y) \times \frac{1}{3}, if\ P - Value\ is\ [0.001, 0.01) \\ 0, o.w. \end{cases} \tag{2}$$

2.3 STRING MATCHING

To match columns with the string data type, three matching scenarios were considered. String match is performed at three stages, including (1) exact match, (2) partial match, and (3) semantic match. In the exact match stage, identical strings were identified between two columns; in the partial match stage, Levenshtein distance, also known as edit distance, was calculated for two strings from distinct columns. Levenshtein distance counts the required steps for one string converted to the other string [3]. Therefore, abbreviations or typos can be recognized in this stage. Finally, WordNet was used in the semantic match stage. WordNet is a large lexical database that stores synsets [4]. Strings with similar meanings can be found in the last stage. Besides, strings would not be compared, when they are matched in the previous stage. The number of pairs of three stages was counted to calculate the similarity score between two columns. The similarity score for column A and column B is calculated as

$$string\ score = \frac{\#exa \times 1.2 + \#par + \#sem \times 1.2}{min(n, m) \times 1.2} \tag{3}$$

where #exa is the number of matches in the exact match stage, #par is the number of matches in the partial match stage, #sem is the number of matches in the semantic match stage, n is the number of distinct elements in column A, and m is the number of distinct elements in column B. Since exact match and semantic match is relatively important, the weighted value was set as 1.2 for the exact and semantic match. The value of the similarity score is [0, 1].

2.4 COLUMN NAMES MATCHING

Word embedding is the mapping of words into a vector space, one kind of Word representation. Word2Vec and FastText, word embedding methods, were used for semantic matching [2]. FastText is a word vector calculation and text classification tool open-sourced by Facebook in 2016 [5]. With FastTsxt fast training and memory-saving features, the number of classes and the amount of data are relatively large and have good results. We utilized the pre-training FastText model for synonym mining and converting column names into vectors. The similarity between word vectors can be calculated as an indicator of matching. The calculation method of similarity adopts cosine similarity. The value of the similarity score is [0, 1].

The last step is to take the weighted average with column name similarity and the numeric or string similarity. The weight is set at default 0.5 and can be adjusted by users.

3 RESULTS

3.1 EXAMPLE: the NBA Player Dataset and the MLB Player Dataset

We use two different but similar datasets to test the matching ability of JOURS. The NBA player dataset[1] and the MLB player dataset[2] are from the Kaggle website. The NBA player dataset is denoted as Source Table, and the MLB player dataset is Test Table. There are 5316 rows and 18 columns in Source Table, and 1034 rows and 8 columns in Test Table. The descriptions of tables are in Table **1** and Table **2**. Besides, the unit of weight and height is different between NBA player data and MLB player data.

Table 1: Column Description of NBA Player Data (Source Table).

Column Name	Data Type
player name	obj
team_abbreviation	obj
age	num
player_height (cm)	num
player_weight (kg)	num
college	obj
country	obj
draft_year	num
gp	num
pts	num
reb	num
ast	num
net_rating	num
oreb_pct	num
usg_pct	num
ts_pct	num
ast_pct	num
season	obj

Table 2: Column Description of MLB Player Dataset (Test Table).

Column Name	Data Type
Name	obj
Team	obj
Position	obj
Height (inches)	num
Weight (pounds)	num
Height (cm)	num
Weight (kg)	num
Age	num

The matching result is stored in Recommend Table (see Table 4). Test col is the column form Test Table, Source Col is the column from Source Table. Recommend Table records the similarity between two columns. The last column is the similarity score of two columns, and it take both column name scores and numeric (or string) score into consideration. The Joinable column means weather two similar columns can be the primary key for combining tables. If the values of two similar columns are unique and there are not missing values in both columns, then the columns may be the primary key.

Table 3 shows each test column and its first matched source column. We can find that overall JUORS can match columns correctly. However, when the two columns should be matched but have different measure units like cm and inches. This might make the match result wrong. In addition, if the two-column should not be matched but have a very similar feature and distribution like age and usg_pct (one kind of basketball performance indicator), the match result will also likely be wrong.

Table 3: Matching Results for the NBA Player Dataset and the MLB Player Dataset.

Test Col	Source Col	Ground Truth	Correct or not
Height (inches)	player_weight	player_height	x
height	player_height	player_height	x
Weight (pounds)	player_weight	player_weight	o
weight	player_weight	player_weight	o
Age	usg_pct	age	x
Name	player_name	player_name	o
Team	team_abbreviation	team_abbreviation	o
Position	team_abbreviation	x	

The analysis was performed by using MacBook Air systems with Apple M1 chip and 8-core GPU, configured with 8GB of RAM and 512GB SSD. The runtime of the analysis is 20.89 seconds which suggested the effectiveness of JOURS.

4 DISCUSSION

The K-means Clustering number is 2 or 3 groups in our default. The main purpose of clustering is to decrease computation time. When the test data table is input into JOURS, it can be converted and categorized into similar groups by the source table clustering method. Therefore, the clustering method and grouping numbers can be modified by prior knowledge or user preference.

Currently, we set that the source table and test table are different. For user-friendly, the two tables can be the same, which often occurs in actual conditions. Also, the JOIN score in our dataset

[1] https://www.kaggle.com/datasets/drgilermo/nba-players-stats

[2] https://www.kaggle.com/c/mlb-player-digital-engagement-forecasting/data

Table 4: Recommend Table for the NBA Player Dataset and the MLB Player Dataset.

Test Col	Source Col	Data Type	Joinable	Overlap Ratio	Colname Score	Numeric Score	String Score	Score
Height (inches)	player_weight	num	none	0.007	0.604	0.000	null	0.302
Height(inches)	player_height	num	none	0.000	0.601	0.000	null	0.301
Height(inches)	ast	num	none	0.000	0.565	0.000	null	0.282
height	player_height	num	none	0.536	0.741	0.000	null	0.371
height	player_weight	num	none	0.000	0.682	0.000	null	0.341
height	ts_pct	num	none	0.000	0.622	0.000	null	0.311
Weight (pounds)	player_weight	num	none	0.000	0.598	0.000	null	0.299
Weight (pounds)	ts_pct	num	none	0.000	0.589	0.000	null	0.295

should be higher than 0.2~0.3. Due to the heterogeneity of datasets, we recommend that the user could adjust the score threshold or the proportion of column names similarity and tokens similarity. In the practical application, the numeric method can be used to determine the unknown column data of the factories.

5 CONCLUSION

In this study, we developed JOURS which is a JOIN and UNION operations recommendation system, and the pipeline of JOURS is described. JOURS implemented schema matching as the mean for data integration. JOURS is a user-friendly tool. Once users import two tables, Source Table and Test Table respectively, in JOURS. Recommend Table is then generated, and it contains the similarity score between two columns from distinct sources. Also, the Recommend Table can be saved in MySQL.

REFERENCES

[1] Jin Wang, Chunbin Lin, and Carlo Zaniolo, "MF-Join: Efficient fuzzy string similarity join with multi-level," *IEEE 35th International Conference on Data Engineering (ICDE),* 2019.

[2] K. Nozaki, T. Hochin, and H. Nomiya, "Semantic schema matching for string attribute with word vectors," *In Proceedings of the 2019 6th International Conference on Computational Science/Intelligence and Applied Informatics (CSII'19),* p. 25–30, 2019.

[3] Li Yujian and Liu Bo, "A normalized Levenshtein distance metric," *IEEE Trans. Patt. Anal. Mach. Intell. 29, 6,* p. 1091–1095, 2007.

[4] G. Miller, "WordNet: A lexical database for English," *Commun. ACM 38, 11,* p. 39–41, 9 1995.

[5] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T, "Fasttext. zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651,* 2016.