ISSN: 2710-0588

A Multi-truth Discovery Algorithm based on Statement Value Grouping and Data Source Information Richness

Dongjun Gao*, Zhiyong Zhang

Information Engineering College, Henan University of Science and Technology, Luoyang, 471023, China

*Corresponding Author

Abstract

As the volume of data continues to grow, it is common for data from the same source to contain multiple domains. Combining domain segmentation can enhance the effectiveness of data fusion. This paper presents a multi-truth discovery algorithm that utilises statement value grouping and domain information richness. The data is first grouped based on their similarity, and the resulting groups replace the original data. Then, the reliability of each data source is calculated by domain. The truth value and reliability of each source are iteratively calculated until the end condition is met. Finally, the appropriate value is selected as the final result from the obtained dataset. Experiments were conducted on real datasets to demonstrate the algorithm's effectiveness.

Keywords

Multi-truth Discovery; Value Similarity; Source Reliability.

1. Introduction

As the Internet becomes more integrated into our daily lives, the amount of data generated by users and devices continues to increase. This often leads to duplicate information, where multiple descriptions of the same object appear when querying for information[1]. Conflicts are almost impossible to avoid when multiple data sources describe the same object due to various reasons, such as incomplete, incorrect, or missing data [2]. Therefore, it is crucial to identify the most comprehensive and reliable information from conflicting or erroneous sources. This process is commonly referred to as the truth discovery problem, which improves data quality and reduces data duplication, resulting in saved storage space[3].

This paper proposes a multi-truth discovery algorithm based on data grouping and reliability initialization of domain data sources. The main contributions are:

(1) Introducing a novel string similarity calculation method to reduce computational complexity by grouping data and eliminating differences between data.

(2) The algorithm's accuracy is improved by refining the calculation process and determining the reliability of the data source in a sub-domain.

The paper's proposed algorithm is proven effective through experiments on real datasets, which utilize domain information to initialize data source reliability.

The paper's remaining sections are structured as follows: Section 2 presents related work, Section 3 outlines the problem, Section 4 details the algorithm's steps, Section 5 presents experimental results, and Section 6 concludes the paper with an outlook.

ISSN: 2710-0588

2. Related Work

Yin et al. proposed the truth discovery problem, which involves finding the most accurate description for each object from conflicting descriptions provided by multiple data sources[4]. The article proposes a single-truth discovery algorithm based on two assumptions regarding the truth discovery problem. Firstly, the reliability of a data source is proportional to the amount of correct information it provides. Secondly, the reliability of a claim value provided by a data source with high reliability is also higher. Based on these two fundamental assumptions, researchers have proposed numerous truth discovery algorithms.

Methods for truth discovery are categorized into two types: single-truth discovery methods [5-6] and multi-truth discovery methods[7-8]. Some methods merge the two[9]. Single-truth discovery methods are used when only one truth value exists for each object, while multi-truth discovery methods are suitable when each object may have multiple truth values. Most algorithms rely on iteration to update the reliability of the true value with respect to the data source until convergence. Table 1 illustrates the multi-truth case, where three data sources (S1, S2, S3) provide information about four books' artistic and literary aspects. This example shows that for book 0030860202, data source S1 provides information for one author, while data source S2 provides information for two authors. The information provided by the two sources does not overlap and conflicts with each other.

Table 1. Book Information Provided by Some Book Websites

Source	ISBN	Category	Authors
S_1	0060196955	Literature	Basbanes Nicholas
	0025486500	Literature	Mitchell Margaret; Harwell Richard Barksdale
	0030860202	Literature	Williams Brad
S_2	0030860202	Literature	Williams, Sandra; Ehrlich, J W
	0847816982	Arts	Collins, Brad; Robbins, Juliette
S_3	0060196955	Literature	Basbanes Nicholas A
	0025486500	Literature	Margaret Mitchell

The probabilistic graph-based model LTM proposed by Zhao et al. was the first method for the multi-truth problem[10]. The method treats the true values as potential random variables, uses Bayesian methods to model the error generation process and the quality of the data source, and finally outputs the values with an estimated probability greater than 0.5 as the true values.The SmartVote method proposed by Fang et al[11] computes various aspects of the reliability of the data source by randomly wandering around the graph and takes into account the phenomenon of the long tail of the data.

3. Problem Description

Table 2. Symbol Definition Table

Symbol	Description
N	Number of data sources
P	Number of entities
D	Number of domains
W	Reliability set of N data sources
w_n^d	Reliability of data source n in domain d
v_p^n	The value of entity p provided by data source n
v_p^*	The true value of entity p
P_d	Collection of entities in domain d
T_n^d	Information richness of data source n in domain d

ISSN: 2710-0588

Assuming N data sources provide claim values for the attributes of p entities in domain D, the claimed values of data sources n for the attributes of entity p are denoted as vpn. The reliability of data sources n in domain d is denoted as wdn. Table 2 illustrates the notations used in this paper and their meanings.

Given a dataset containing D domains, N data sources, and P entities, each with at least one data source providing multiple declared values for its attributes, the goal is to determine the true value of entity P from the declared values provided by multiple data sources in conjunction with the reliability of the data sources, with the aim of minimizing the deviation between the true value and the declared value. The problem can be expressed mathematically as finding the minimum value of the following equation:

$$\min_{v^*,W} R(V^*,W) = \sum_{d=1}^{D}\sum_{n=1}^{N} w_n^d \sum_{p=1}^{P} d(v_p^*, v_p^n) \tag{1}$$

where $d(\bullet)$ denotes the distance between the claimed value and the true value.

$$d(v_p^*, v_p^n) = \begin{cases} 1, & v_p^* \neq v_p^n \\ 0, & v_p^* = v_p^n \end{cases} \tag{2}$$

4. Algorithm Description

The overall flow of the algorithm in this paper is shown in Figure 1.

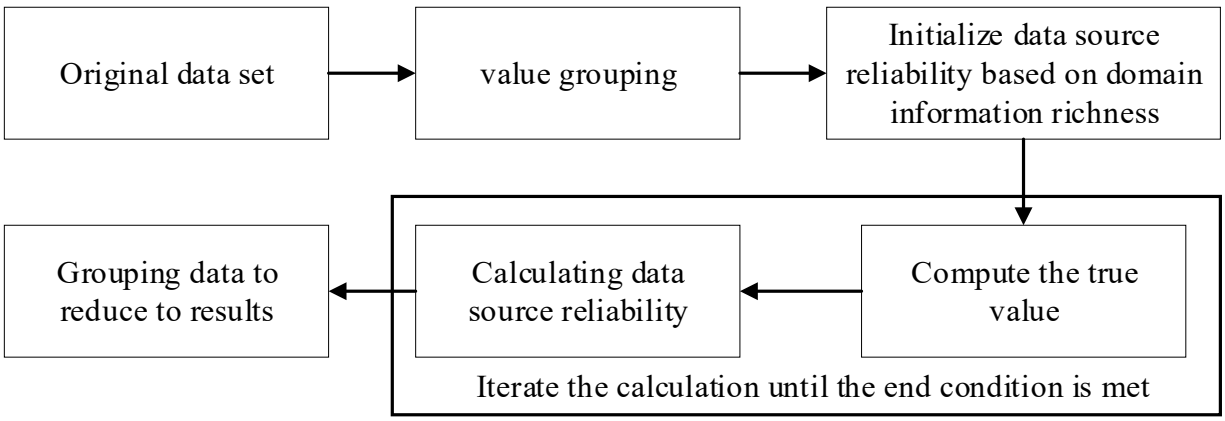


Figure 1. Two or more references

4.1. Processing of Claim Values

Table 3. Part of the Author Information Data of the Book Sea of Glass (ISBN: 0312007809)

Book	Author
Sea Of Glass	Longyear, Barry B.
Sea Of Glass	LONGYEAR, Barry
Sea Of Glass	Barry B. Longyear
Sea Of Glass	Longyear
Sea Of Glass	Longyear, Barry
Sea Of Glass	Barry B Longyear
Sea Of Glass	Barry Longyear

ISSN: 2710-0588

To better illustrate the necessity of the claim value processing step and the challenges it poses, we explain the problem using a real book dataset, which is also the dataset used in the experimental part. The author information of the book "Sea Of Glass" in this dataset is randomly selected for analysis, and the partial data provided in the dataset about the author attributes of the book are shown in Table 3. According to the data in Table 3, we can see that although the data in the table are in different formats, they are actually just different expressions of the same person's name (Barry B. Longyear). Eliminating the differences in them through the claim value processing process can improve the accuracy of the subsequent truth discovery process.

Looking at the author information in Table 3, we can see that the differences that exist in the author information come from two main sources:

(1) Simple differences. Differences caused by the use of different cases for the same word, such as: all uppercase, all lowercase, and mixed case. Different punctuation is used to separate or abbreviate names, and names are written in a different order, such as last name first or first name first.

(2) Complex differences. Some data sources use abbreviated forms of names, omit middle names, and misspell letters in some names.

For the above two types of discrepancies, we design an appropriate scheme to eliminate the discrepancies. For simple differences, we design a simple data cleaning scheme to standardize the format and output the corresponding data set. For the second type of more complex differences, we use a new string similarity metric algorithm to determine the relationship between strings, and collect strings with the same metric results into the same group, and use the number of this group to replace the string in the next calculation.

1.1.1. Clean Data

The steps of the methodology used to eliminate simple inconsistencies and harmonize data formats are as follows:

(1) Remove all numbers and redundant punctuation, such as separators after initials and commas between names.

(2) Harmonize case by converting all characters to all lowercase.

(3) Solve the problem of different order of names by separating the tokens that make up the string and reordering them alphabetically.

1.1.2. String Similarity Measure

A string similarity measure is performed on the cleaned data from the previous step, and strings with similarity measures greater than a specified value are grouped together.

For the classical similarity measures, Hamming, Levenshtein, Jaro-Winkler, Jaccard, Sørensen, Ratcliff-Obershelp, etc. were tested and found that these algorithms are not a good solution to the problem of similarity measures in this paper. For example:

(1) For 'longyear' and 'barry longyear', two strings with the same meaning, several of the above algorithms give similarity results less than 0.5.

(2) For two different values, 'clive longyear' and 'barry longyear', these algorithms again give results greater than 0.5.

Based on the above examples, we can see that these classical string similarity measures do not give the desired results in the case of this paper. Therefore, in this paper, we use a novel similarity measure for string similarity to eliminate the string differences caused by abbreviations as well as spelling errors. The algorithm inputs two strings and outputs the similarity of these two strings.

First, a similarity matrix M is constructed in which the elements in each cell m_{ij} represent the similarity between the ith word of the first string and the jth word of the second string. Where

ISSN: 2710-0588

word similarity is defined as the percentage of the length of the longest prefix match to the shorter of the two words.

$$SL = \frac{Maximum\ match\ prefix\ length}{\min(l_1, l_2)} \tag{3}$$

Example 1: Compute the similarity matrix of the strings 'b bullen j' and 'barrie bulle' from the above description. $m_{11} = 1/1 = 1$, $m_{12} = 1/6 = 0.16$.

$$\begin{matrix} & b & bullen & j & \\ \mathrm{M} = & \begin{bmatrix} 1 & 0.16 & 0 \\ 1 & 1 & 0 \end{bmatrix} & \begin{matrix} barrie \\ bullen \end{matrix} \end{matrix} \tag{4}$$

In example 1, we can see that a word in the first string can have a similar relationship with more than one word in the second string, so we need to avoid this one-to-many matching when summarizing the overall similarity of this matrix, i.e. we can only select one value for each row and column of the similarity matrix. The calculation of the matrix similarity matching score can then be expressed in the following mathematical form:

$$G = \sum_{i=1}^{l_i} \sum_{j=1}^{l_j} m_{ij} p_{ij} \tag{5}$$

where l_i and l_j are the number of words in the first and second strings, respectively. p_{ij} is a variable indicating whether the element in row i and column j of the matrix is selected, if the element is selected then $p_{ij} = 1$, otherwise $p_{ij} = 0$.

And the following constraints must be met:

1. $\sum_{i=1}^{l_i} m_{ij} = 1, j = 1, \ldots, l_j$: Each word in the second string can match only one word in the first string.
2. $\sum_{j=1}^{l_j} m_{ij} = 1, i = 1, \ldots, l_i$: Each word in the first string can match only one word in the second string.

In this section, the Hungarian algorithm is used to compute the optimal solution to the problem. The similarity matrix is input to the algorithm and the algorithm returns the maximum value of this similarity matrix, which is the sum of the similarities between the two strings. The return value is divided by the average number of words in the two strings as the final similarity result. If the result of $Sim(s_1, s_2)$ is greater than 0.5, we consider the two strings to be similar, otherwise it is noted that the two strings are different.

$$Sim(s_1, s_2) = \frac{G}{avg(l_1, l_2)} \tag{6}$$

ISSN: 2710-0588

Table 4. String similarity algorithm

Algorithm 1 Similarity Algorithm
Input: string s_1, s_2
Output: similarity of strings s_1, s_2
$c_1 \leftarrow \lvert s_1 \rvert$
$c_2 \leftarrow \lvert s_2 \rvert$
$M \leftarrow [c_1][c_2]$
for all $word_i \in s_1$ **do**
for all $word_j \in s_2$ **do**
$m_{ij} = SL(word_i, woed_j)$
$G = Hungarian(M)$
$Sim(s_1, s_2) = G / avg(l_1, l_2)$
return $Sim(s_1, s_2)$

4.2. Data Grouping

Table 5. Value Grouping Algorithm

Algorithm 2 Value Grouping Algorithm
Input: Entity data in domain d
Output: Entity data in domain d after attribute encoding
for all $p \in P_d$ **do**
$V(p) \leftarrow$ N data sources provide values for entity p
$G(p) \leftarrow \varnothing$
while $V(p) \neq \varnothing$ **do**
$mark \leftarrow MostFrequent(V(p))$
$V(p) \leftarrow V(p) \setminus \{mark\}$
$group \leftarrow \{mark\}$
for all $v \in V(p)$ **do**
if $Sim(mark, v) > 0.5$ **do**
$group \leftarrow group \cup \{v\}$
$V(p) \leftarrow V(p) \setminus \{v\}$
$G(p) \leftarrow G(p) \cup \{group\}$
for all $r \in r(p)$ **do**
for all $str \in r$ **do**
for all $group \in G(p)$ **do**
if $str \in group$ do
$str \leftarrow group$
return I_d

After completing the appropriate string similarity metric algorithm, we can aggregate strings with the same meaning into a group and use the grouped results instead of the original strings and use the grouped results in the next computation process.

ISSN: 2710-0588

The data grouping algorithm first selects the value with the highest number of occurrences from the set of values provided by all data sources for a particular object, creates a group containing that value, and numbers it.

Next, iterates through the remaining values in the set, selects all values with a similarity greater than 0.5 to the current value to add to the current group, and removes the values added to the current group from the set of all values. Then it selects the value with the highest number of occurrences from the remaining set of values and continues the previous steps until the iteration stops when the set of values is empty.

Finally, the values of all objects are replaced with the corresponding value group numbers.

4.3. Data Source Reliability Initialization

Obviously, it is not reasonable to use the average value to initialize the reliability of a data source, and in this paper we will consider the difference in the contribution of the data source in each domain to initialize its reliability in a reasonable way.

The richness of data provided by different data sources in different domains can be very different. For example, if a data source provides a large amount of information about children's books, but only a small portion of science and technology books, it is more likely that the data source has a higher reliability in children's books. Therefore, it is necessary to initialize the reliability of the data source according to the richness of the domain information provided by the data source. The ratio of the amount of data provided by data source n to the amount of all data in domain d is called the domain information richness of the data source, which is calculated as follows:

$$T_n^d = \frac{\left|O_n^d\right|}{\sum_{n\in N}\left|O_n^d\right|} \tag{7}$$

Where $\left|O_n^d\right|$ is the amount of data provided by data source n in domain d.

Based on the domain richness of the data sources, we can initialize the data sources according to the following formula. Where α is a predefined adjustment factor that is used to amplify the difference in reliability between individual data sources.

$$w_n^d = \alpha \cdot (1 + T_n^d) \tag{8}$$

4.4. Updating the Truth Value

After the reliability of the data source is initialized, the truth value is calculated using the result of the initialization according to the following formula. Find the value that minimizes the distance between the claimed value and the true value as the current true value.

$$v_i^* \leftarrow \arg\min \sum_{n=1}^{N} w_n^d * d(v^*, v_n^i) \tag{9}$$

4.5. Updating Data Source Reliability

The reliability of a data source is updated after the truth value is obtained. The reliability of a data source is expressed as the ratio of the distance between the claimed value and the true value provided by the data source n to the sum of the distances between the claimed value and the true value of all data sources in the domain. The greater the distance between the claimed value and the true value provided by the data source, the lower the reliability of the data source. The specific calculation formula is:

ISSN: 2710-0588

$$w_n^d = \frac{\sum_{p=1}^{P_d} d(v_p^*, v_p^n)}{\sum_{n=1}^{N} \sum_{p=1}^{P_d} d(v_p^*, v_p^n)} \tag{10}$$

where T is an object whose data source n provides a claimed value in domain d.

4.6. Selecting the Truth Value

After iteratively updating the truth value with the reliability of the data source until the loop ends, if the constraints are satisfied. At the end of the loop, the output truth value is the number of the encoded data group. To restore the numbering to the form of the original data, we select the string with the highest number of occurrences in each group as the final output truth value.

Table 6. VDTD Algorithm

Algorithm 3 VDTD
Input: Original data set I
Output: Truth dataset I_1
1: **while** Failure to meet closing conditions **do**
2: **for** all $d \in D$ **do**
3: $I_d \leftarrow Value\_grouping(I_d)$
4: Calculate the domain information richness of the data source according to eq.7
5: Initialize data source reliability according to eq.8
6: **for** all $p \in P_d$ **do**
7: Calculation of v_p^* according to eq.9
8: Calculation of w_n^d according to eq.10
9: $I_1 \leftarrow$ select the value with the highest number of occurrences in the attribute group as the true value
10: **return** I_1

5. Experiments

5.1. Experimental Environment

All algorithmic codes in this paper are written in the Python language. The specific configuration of the experimental environment is as follows:

(1) Hardware environment: Intel(R) Core(TM) i5-10500 CPU; 16 GB RAM.

(2) Software environment: Windows 11; Python 3.7.

5.2. Dataset

Book[12]: This dataset is a real world dataset. The dataset contains 54591 different book information providers providing 2338559 book author information for 210206 books from 18 domains. Each of these book information providers provided information for at least one book. In addition, we randomly selected 395 books and manually recorded the author information on the book cover as the true value of the book's author information.

ISSN: 2710-0588

5.3. Comparison Methods

Voting: a majority voting method that takes the value with the highest number of occurrences in the declared value of the data source as the true value.

Truth Finder[4]: uses an iterative approach to compute data source reliability and attribute truth values based on two basic assumptions of truth discovery.

CATD[13]: improves the accuracy of estimating the reliability of a data source by taking into account the long-tail phenomenon of the data and reliability confidence intervals, and obtains the true value.

5.4. Experimental Results

The information of the dataset obtained after following the steps of the algorithms in Section 4.1.1 for the original dataset and deleting the data without authors as well as the data of the books where all data sources give the same information about the authors is shown in Table 7. All subsequent algorithms are computed on the cleaned dataset. The data in the cleaned dataset is significantly reduced by removing most of the non-conflicting data information, which reduces the subsequent computational overhead. Figure 2 shows the top 10 data sources in terms of data volume in the cleaned dataset in the domains Business, Science Fiction, Literature, Biographies, Arts, Romance, Social Sciences, Cookbooks, History, Religion, as well as their share in the original dataset.

Table 7. Comparison of Information Before and After Data Set Cleaning

	Number of entities	Number of data sources	Number of records
Before Cleanup	210206	54591	2338559
After Cleanup	94745	6533	2134858

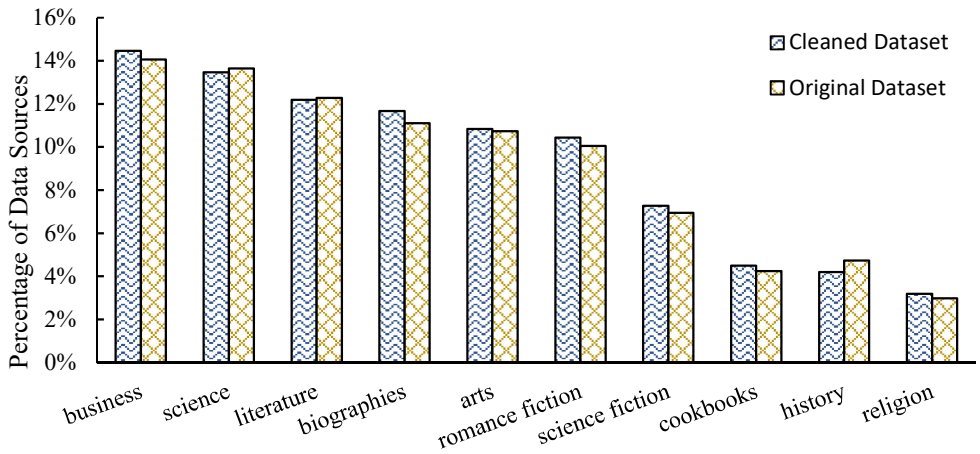


Figure 2. Proportion of Data in Each Domain

Figure 3 shows the information richness of the five booksellers in the cleaned dataset in the domains religion, science fiction, self-help, social science, and travel.Wonder Book and Revaluation Books for example, we can see that in the domains religion and We can see that in the domains religion and science fiction, Wonder Book provides more data, but in the remaining three domains, Revaluation Books provides more information. It can be seen that domain segmentation of data can improve the degree of variation between data sources and is more reasonable compared to the average reliability of data sources.

ISSN: 2710-0588

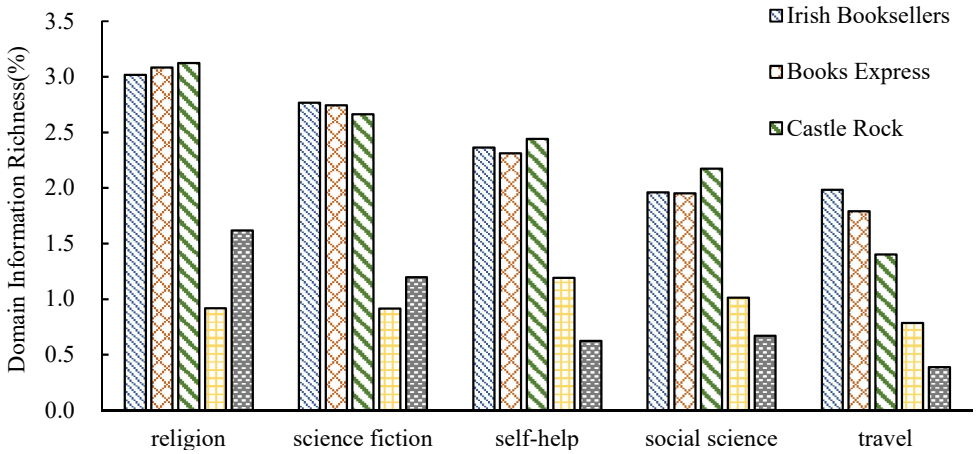
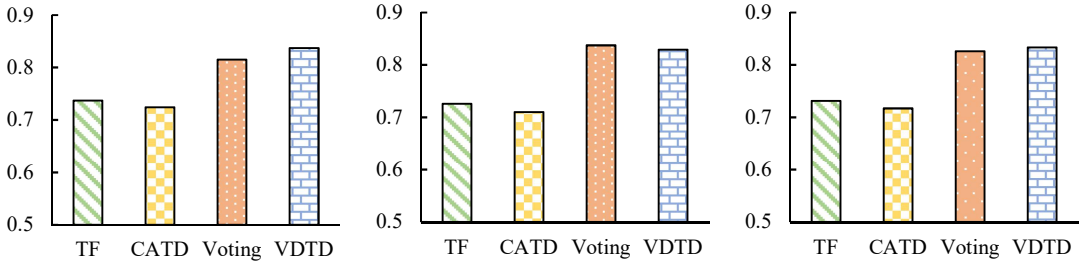


Figure 3. Domain Information Richness of Some Booksellers in the Cleaned Data Set



(a) Precision (b) Recall (c) F1

Figure 4. Experimental Results of the Algorithm on Book Data Set

Table 8. Experimental Results of the Algorithm on Book Data Set

Method	Precision	Recall	F1
TF	0.73705	0.72549	0.73123
CATD	0.72399	0.70980	0.71683
Voting	0.81489	0.83725	0.82592
VDTD	0.83725	0.82913	0.83317

In this paper, precision, recall, and F1 are used as evaluation metrics for the algorithm. Where precision indicates the proportion of results obtained by the algorithm that are equal to the true value, recall indicates the proportion of results obtained by the algorithm that are correct as a percentage of the true value, and F1 is the reconciled mean of precision and recall.

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{11}$$

Table 8 shows the results of all the algorithms run on the book dataset. The result that can be clearly observed from Table 8 is that the F1 score of the VDTD algorithm has the best result among all the algorithms compared, which achieves a better balanced result between precision and recall. Secondly, from the results in the table, it can be seen that the algorithms TF and CATD for single-truth have worse results because most of the books in this dataset have multiple authors and the algorithms for single-truth obviously do not deal with this problem very well. the Voting algorithm has relatively better results because basically there are multiple records for each book in this dataset to provide the authorship information for it, so the number of occurrences of them are likely to be true values. The VDTD algorithm proposed in this paper

ISSN: 2710-0588

divides the domain of the data so that the values provided by the data sources with higher information richness in the domain are more informative, thus avoiding the errors caused by the wrong data provided by some data sources with lower domain richness and improving the accuracy of the algorithm.

6. Conclusion

In this paper, a multi-truth discovery algorithm, VDTD, is proposed, which first groups the data in conjunction with a novel character similarity metric, second initializes the reliability of the data source in conjunction with its domain information richness, then iteratively updates the truth value and the reliability of the data source, and finally selects the appropriate content as the truth value from the resulting grouping. The algorithm in this paper shows better F1 values than the remaining three algorithms on real data sets, demonstrating the effectiveness of the algorithm. Since the inference of the reliability of the data source in this algorithm mainly relies on the amount of information provided by the data source, in the future, we can consider how to adjust the domain richness of the data source in a timely manner for the dynamic increase in the amount of data.

References

[1] Sun P, Wang Z, Wu L, et al. Towards Personalized Privacy-Preserving Incentive for Truth Discovery in Mobile Crowdsensing Systems, IEEE Transactions on Mobile Computing, Vol. 21 (2020) No.1, p.352-365.

[2] Tang J, Fu S, Liu X, et al. Achieving Privacy-Preserving and Lightweight Truth Discovery in Mobile Crowdsensing, IEEE Transactions on Knowledge and Data Engineering, Vol. 34 (2021) No.11, p.5140-5153.

[3] Li Y, Sun H, Wang W H. Towards Fair Truth Discovery from Biased Crowdsourced Answers, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2020).p.599-607.

[4] Yin X, Han J, Yu P S. Truth Discovery with Multiple Conflicting Information Providers on the Web, IEEE Transactions on Knowledge & Data Engineering, Vol. 20 (2008) No.6, p.796-808.

[5] Lyu S S, Ouyang W T, Wang Y Q, et al. Truth Discovery by Claim and Source Embedding, IEEE Transactions on Knowledge and Data Engineering, Vol. 33 (2021) No.3, p.1264-1275.

[6] Ye C, Li Q, Zhang H, et al. Autorepair: An Automatic Repairing Approach over Multi-source Data, Knowledge and Information Systems, Vol. 61 (2019), p.227-257.

[7] Chang C, CAO J J, Feng Q, et al. Truth Discovery of Multi-source Text Data, IEICE Transactions on Information and Systems, Vol. 102 (2019) No.11, p.2249-2252.

[8] Fang X S, Sheng Q Z, Wang X Z, et al. From Appearance to Essence: Comparing Truth Discovery Methods Without Using Ground Truth, ACM Transactions on Intelligent Systems and Technology, Vol. 11 (2020) No.6, p.1-24.

[9] Xu Y, Cao J J, Weng N F, et al. An Adaptive Truth Discovery Model Consisting of Multiple Truth Part and Single Truth Part, 2021 International Conference on Cyber-Physical Social Intelligence (2021). p.1-5.

[10] Zhao B, Rubinstein B I P, Gemmell J, et al. A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration, Proceedings of the VLDB Endowment, Vol. 5 (2012) No.6, p.550-561.

[11] Fang X S, Sheng Q Z, Wang X Z, et al. Smartvote: A Full-Fledged Graph-Based Model for Multi-valued Truth Discovery, World Wide Web, Vol. 22 (2019) No.4, p.1855-1885.

[12] Lin X, Chen L. Domain-aware multi-truth discovery from conflicting sources, Proceedings of the VLDB Endowment, Vol. 11 (2018) No.5, p.635-647.

[13] Li Q, Li Y, Gao J, et al. A Confidence-Aware Approach for Truth Discovery on Long-Tail Data, Proceedings of the VLDB Endowment, Vol. 8 (2014) No.4, p.425-436.