IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 12, 2020, accepted December 23, 2020, date of publication December 29, 2020,
date of current version January 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047927

String Comparators for Chinese-Characters-Based Record Linkages

SENLIN XU, MINGFAN ZHENG, AND XINRAN LI
Department of Mathematics and Statistics, College of Science, Huazhong Agricultural University, Wuhan 430070, China
Corresponding author: Xinran Li (xinran.li@mail.hzau.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 81701794 and in part by the Fundamental Research Funds for the Central Universities under Grant 2662018QD010.

ABSTRACT In the context of big data, data sharing between different institutions can not only reduce the cost of information collection greatly but also benefit for obtaining analysis results effectively and efficiently. Record linkage is the task of locating records that refer to the same entity from heterogeneous data sources. In the last decades, extensive researches on alphabet-based record linkages have been carried out, among which the Fellegi-Sunter model extended by Winkler has outperformed others. However, it is still a challenge to perform record linkage on Chinese-character-based datasets. In this article, two set-based methods (Cosine similarity and Dice similarity) were introduced firstly, and then the similarity of Chinese characters was quantified based on an adapted encoding technique which exploits the information of both the shape and the pronunciation of Chinese character. A new method entitled Hybrid similarity was proposed in the next part, which is the combination of the character transformation technique (SoundShape Code) and Dice similarity. Finally, we performed the aforementioned methods on the simulated datasets, and each method was evaluated by counting the number of misclassified record pairs and the computational time. The results demonstrated that our Hybrid similarity method outperformed others in reducing the number of misclassified pairs with a relatively low computational cost.

INDEX TERMS Record linkage, Chinese characters, soundshape code, string comparator, Fellegi-Sunter model.

I. INTRODUCTION

With the development of information technology, data sharing has become more and more important for both enterprises and governments [1]. In the process of data sharing, records referring to the same entity from different databases always need to be linked. However, it is not easy to link the records due to the lack of a unique identifier (UID) between different databases and the heterogeneity of data. Therefore, to determine whether the records refer to the same entity, approximate matching techniques for record linkage have to be proposed to compare the corresponding fields (such as name, birthdate, address, etc.) in different records.

The simplest approach for linkage is what we shall call deterministic record linkage, which requires the perfect agreement of all or a predetermined set of fields between two records to consider them as belonging to the same entity [2]. The method for deterministic record linkage is appropriate when the data quality of identifier fields in the databases to link is relatively high [3]. But identifier fields are sometimes subject to misspellings and typographical errors [4]–[6]. In addition, some fields provide more information more reliable than others [7]. For example, two records whose name field agreement are much more likely to refer to the same person than those whose address field agreement. However, compared with the probabilistic record linkage, the deterministic record linkage does not make such a distinction [8], [9].

The commonly used probabilistic record linkage was formalized by Fellegi and Sunter (referred to as PRL-FS in this article) [10]. As for this method, each field is assigned with an agreement weight and a disagreement weight based on the log-likelihood ratios. For each record pair, the weight is computed by summing all fields' weights. Specifically, when a field agrees, its field agreement weight is used for the record pair's weight computation; otherwise, its field disagreement weight is used. To make a linkage decision, we can compare record pairs' weights with a decision threshold, above which record pairs can be considered as matches, and below which

The associate editor coordinating the review of this manuscript and approving it for publication was Mingjun Dai.

TABLE 1. Examples for Chinese Characters Based Records.

Database	Name	Address
File A	许艺	湖北省武汉市
	王轶	广东省广州市
File B	徐艺	湖北省武汉市洪山区
	王秩	广州市白云区

record pairs can be considered as non-matches [11]. The PRL-FS method is relatively simple to implement, but it has a disadvantage of assigning consistent weights to two identical strings, and not assigning inconsistency weights to the similarity between two strings. Winkler therefore proposed an enhanced PRL-FS method (referred to as PRL-W) considering field similarity in the field weights computation and proved its outperformance over the PRL-FS [12].

The above-mentioned methods for record linkage were initially proposed and applied to datasets in alphabet-based languages, and the similarity measurements for strings being composed of letters have been effectively solved and wildly applied. However, the application of similarity measures based on Chinese characters still has its limitation in the task of record linkage. Unlike the alphabet-based languages, Chinese characters are ideographs (a written symbol that represents a meaning directly rather than a speech sound). There are more than 70,000 Chinese characters in the linguistics dictionary, including about 3,500 of the most commonly used Chinese characters [13], many of which have the same/similar pronunciation and/or similar shape. Therefore, errors in the records may sometimes occur during the process of input, transcription, or OCR recognition [14]. For example, the Chinese surnames (as shown in Table 1) "许" and "徐" respectively in names "许艺" and "徐艺" have the similar pronunciation "xu" but with different tones; while the first names "轶(yi)" and "秩(zhi)" respectively in names "王轶" and "王秩" have a similar shape but with different pronunciations.

In representations of Chinese characters stored, none of the above complexities are directly encoded in computer memory. Therefore, naively applying the traditional string similarity method to the Chinese character field will not be able to solve the errors correctly in the records, which leads to poor ability to discriminate between co-referent and non-co-referent strings [15]. In this article, we extended the existing similarity measures of Chinese characters and proposed a novel string comparator for Chinese-characters-based record linkage.

A. CONTRIBUTIONS

We adapted an encoding technique which exploit the information of both shape and pronunciation of Chinese characters, and based on which, we quantified the similarity for Chinese characters. A novel computational method was proposed to measure the similarity of Chinese names, which considers the shape, pronunciation, and order of characters in two compared strings. We evaluated comprehensively our approach with synthetic datasets, and confirmed the outperformance of this method.

B. OUTLINE

The remaining part of this article is organized as follows: we present related work to measuring the similarity of Chinese characters in Section 2, and in the next section, the classification of PRL-W and our proposed methods for calculating the similarity of two Chinese strings are described in detail. In Section 4, we first provide a process of generating synthetic datasets with different types of errors in records. Then, four methods are evaluated by (A) the performance measures of precision, recall, and f1-score, (B) the numbers of misclassified record pairs and (C) their computational time. And Section 5 is the conclusion of the article.

II. RELATED WORK

At present, there are a few theories focused on the measurement of the similarity of Chinese strings. The initial one is to apply the paired comparison methods (such as Cosine similarity and Dice similarity) to the Chinese string. This method is computationally inexpensive and performs well in long texts. However, in record linkage, since the fields to be compared are usually short texts, resulting in a very sparse distribution of similarities, which make the above similarity measures not appropriate to learn the component similarity of Chinese strings. Therefore, a partial solution to measuring string similarity for logographic scripts is to encode the original logograms in formats that represent their phonetic or visual properties, or keystroke input sequences, before applying pairwise comparison methods, such as Levenshtein distance [15].

Liu and Lin [16] proposed methods for identifying visually similar Chinese characters by adopting and extending the basic concepts of the Cangjie method (a Chinese input method which defined the 24 basic elements of a Chinese character and a series of rules to decompose the character into these basic elements.). However, they did not explicitly mention the similarity measurement algorithm based on the encoding method. Song *et al.* [17] then improved an algorithm to measure Chinese character similarity based on structural information. The algorithm first decomposed a Chinese character into several smaller components and calculated the similarity score between the compared components, and the final score was given by the weighted average of the similarity scores between the components. Chang *et al.* [18] designed a method based on simple rules to measure the phonetic similarity between two Chinese characters, where the Mandarin phonetic symbols of two characters are compared respectively. Ming Liu *et al.* [19] then proposed an improved approach to encode simplified Chinese characters from the perspective of character shape, pronunciation, and meaning, which is used as the basis to calculate the similarity of Chinese characters.

Chen *et al.* [20] proposed a method for converting a Chinese character to SoundShape Code, which considered the characteristics of Chinese characters in terms of their glyphs and sounds. The score between the SoundShape Code calculated based on the Hanming distance was then used as the similarity between the target characters. Similarly, H. Wang *et al.* [21] proposed a method for measuring the similarity of Chinese strings based on SoundShape Code. By converting the target words into SoundShape Code and using an improved editing distance algorithm to measure the similarity between the compared strings. Collender *et al.* [15] developed a framework leveraging multiple comparison features of encoded logographic names to recover aspects of phonetic, visual, and keystroke similarity using machine learning classifiers. However, this method requires a large amount of annotated data.

III. METHODS

In this section, we first describe a commonly used parameters estimation method for record linkage based on EM (Expectation Maximization) Algorithm. Then, we introduce two set-based string comparison algorithms: Cosine similarity and Dice similarity [22], and how they can be used to compare the content of fields in a record pair. Considering the limitation of the above two methods, we thus proposed an improved Chinese string comparison function named Hybrid similarity (especially for name matching), which is the enhanced version of Dice similarity.

A. METHOD OF CLASSIFICATION OF PRL-W

As an extension of Fellegi and Sunter approach (PRL-FS), the PRL-W method considers a partial level of agreement where the values of string comparator are broken out as different non-intersecting subintervals of [0, 1]. The m-probability $m_{i,s}$ is the conditional probability that the similarity of field i within a record pair falls in the interval s given that the record pair refers the same entity, and the u-probability $u_{i,s}$ is the conditional probability that the similarity of field i within a record pair falls in the interval s given that the record pair refers different entities. Calculating the conditional probabilities defined above is an important aspect of the probabilistic record linkage approach. These probabilities can be obtained by manually assessing the quality of the databases to be matched, or by manually assessing prior matches to the same database. In this article, we estimate the values of these two types of probabilities by employing the EM algorithm [23]. Then, based on m-probabilities and u-probabilities, the weight for the similarity of field i falls in the interval s is calculated as:

$$w_{i,s} = log_2(m_{i,s}/u_{i,s}). \tag{1}$$

and the weight for a record pair is computed by summing all field's weights:

$$w = \sum_i w_{i,s}\mathbb{I}(\gamma_i \in s). \tag{2}$$

where γ_i is the similarity of the record pair in field i.

TABLE 2. Example for PRL-W Method.

Item	Name	Address	Address
Record Pair	许艺	湖北省武汉市	M
	徐艺	湖北省武汉市	M
Interval	[0.8, 1)	1	1
Weight	12.44	10.32	0.98

For example, assume that the two records to compare are given in Table 2, where the similarity of Name falls in the interval [0.8, 1) and the Address and the Sex are identical. Therefore, computed by using (2), the overall weight for the record pair is $12.44 + 10.32 + 0.98 = 23.74$, which is higher than a predefined threshold, meaning that the above two records are more likely to refer to the same entity.

Finally, using the estimated value of the parameter p (the proportion of record pairs involving the same entity), a threshold-based decision rule can be obtained:

The record pair is considered as match if $w \geq T_C$; otherwise, the record pair is considered as non-match. where the threshold T_C is the p^{th} quantile of the weights of all record pairs in descending order [23].

B. EXISTING SIMILARITY MEASUREMENT FUNCTIONS

1) COSINE SIMILARITY

Cosine similarity was originally used to calculate the angle between two vectors in high-dimensional space. We apply this idea to field matching, assuming that for two given records, there are n unique characters in a field, and $v_a = [f_1^a, f_2^a, \cdots, f_n^a]$ is the word vector of record r_k and $v_b = [f_1^b, f_2^b, \cdots, f_n^b]$ is the word vector of record r_l, where f_n^a is the term frequency. Then the Cosine similarity is calculated as:

$$Cosine(a, b) = \frac{1}{||v_a||_2 \cdot ||v_b||_2} \sum_{m=1}^{n} f_m^a \cdot f_m^b. \tag{3}$$

where a and b are the contents in the field of the two records.

$$||v_a||_2 = \sqrt{\sum_{m=1}^{n} (f_m^a)^2}, \quad ||v_b||_2 = \sqrt{\sum_{m=1}^{n} (f_m^b)^2}.$$

As an example, let $a =$ "许艺" and $b =$ "徐艺", then the corresponding word vectors would be $v_a = [1, 0, 1]$, $v_b = [0, 1, 1]$, and the corresponding Cosine similarity will be:

$$Cosine(a, b) = \frac{1}{\sqrt{2} \times \sqrt{2}}$$

2) DICE SIMILARITY

Another commonly used metric of set-based similarity is the Dice coefficient, with which we can measure the similarity by calculating the ratio of common characters contained in two strings. For two given strings a and b, we can calculate their

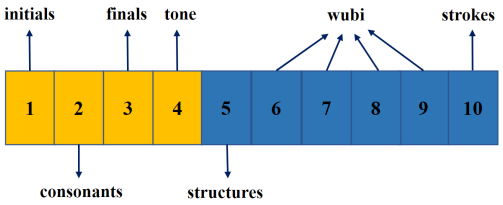


FIGURE 1. The composition of SoundShape Code (SSC).

Dice similarity as follows:

$$Dice(a, b) = \frac{2c}{|a| + |b|}. \tag{4}$$

where c is the number of common characters contained in a and b, $|a|$ and $|b|$ are the length of string a and b. For example, the Dice similarity between "许艺" and "徐艺" will be 0.5.

C. PROPOSED METHOD OF HYBRID SIMILARITY

However, the two set-based similarity functions abovementioned do not take the degree of similarity between two Chinese characters into consideration. This often results in a low similarity score for the compared fields, especially for Chinese names. Therefore, we have proposed an improved Dice similarity computation algorithm named Hybrid similarity.

1) SoundShape CODE

To convert Chinese characters into codes, we first adapted an encoding method for Chinese characters called SoundShape Code (referred to as SSC) [20], [21]. The main idea of SSC is encoding Chinese characters according to their pronunciation and shape into a string (with a fixed length of 10) mixed of letters and numbers. As shown in Fig. 1, the first 4 positions constitute the sound code, while the last 6 positions constitute the shape code, and each part is briefly introduced below.

As shown in Fig.1, the sound code part of an SSC consists of the initials, consonants, finals, and tones of pinyin (a system for romanizing Chinese ideograms in which tones are indicated by diacritics and unaspirated consonants are transcribed as voiced [24]). The shape part of an SSC consists of structure, WuBi (an input method decomposing the Chinese characters into letters combinations [25]), and the number of a stroke to construct a Chinese character. One adaptation is that we replaced four-corner-coding with WuBi-coding in the original SSC [21], since that WuBi-coding is more commonly used in Chinese input methods and contains more information about glyphs.

Considering the pronunciation habits of Chinese characters, we apply the rules given in Table 3 [21] to transform the initials (the first position in SSC) into a number or a letter. For example, the initials "sh" and "s", "zh" and "z", and "ch" and "c" have respectively the same code "G", "D" and "F" because their pronunciations are often confused by people from certain regions.

TABLE 3. Conversion for Initials.

Initials	Code	Initials	Code	Initials	Code	Initials	Code
b	1	f	2	m	3	f	4
d	5	t	6	n	7	l	7
g	8	k	9	h	A	j	B
q	C	x	D	zh	E	ch	F
sh	G	r	H	z	E	c	F
s	G	y	I	w	J		

TABLE 4. Conversion for Finals/Consonants.

Finals	Code	Finals	Code	Finals	Code	Finals	Code
a	1	o	2	e	3	i	4
u	5	v	6	ai	7	ei	7
ui	8	ao	9	ou	A	iu	B
ie	C	ve	D	er	E	an	F
en	G	in	H	un	I	ven	J
ang	F	eng	I	ing	H	ong	K

Analogously, we use a corresponding table (Table 4 [21]) for finals and consonants (if there is no consonant in a pinyin, we use '0' as a placeholder). For example, the finals "an" and "ang" have the same code "F" because it is not easy to distinguish the two for some people in certain areas. In addition, based on the pinyin system, the pronunciations of many Chinese characters have the same initials and finals, but with 4 different kinds of diacritics denoting tones, indicated as 1, 2, 3, and 4 here in this article.

As for strokes, the following rules are employed:

if $1 \leq n_s \leq 9$, we adopt the real number of strokes in a character;

if $10 \leq n_s \leq 35$, we convert them to alphabet: 10 to 'a', 11 to 'b', . . . , 35 to 'z';

if $n_s > 35$, we label it as '0'. where n_s is the number of strokes constituting a character.

Finally, the conversions for WuBi and structure of characters are given in corresponding tables we have collected [26], [27].

2) HYBRID SIMILARITY BASED ON SSC AND DICE

Based on the technology of sound-shape code (SSC), we can convert a Chinese character into a 10-letter/number code. The similarity between two Chinese characters can therefore be measured by comparing their corresponding SSCs. For two given characters e and e', we first convert them to SSCs, $ssc_e = [e_1, e_2, \cdots, e_{10}]$ and $ssc_{e'} = [e'_1, e'_2, \cdots, e'_{10}]$, then we can compute the weighted average of their bit-wise comparisons as SSCS (similarity of SSC):

$$SSCS(e, e') = \sum_{k=1}^{10} w_k \mathbb{I}(e_k = e'_k). \tag{5}$$

TABLE 5. Examples for Encoded Characters.

Characters	SSC	SSCS
许	D6031YTFH6	0.6
徐	D6021TWTYA	
轶	I4041LRWY9	0.6
秩	E4041TRWYA	

where $[w_1, w_2, \cdots, w_{10}]$ are the weights that we preset to $[0.2, 0.05, 0.2, 0.05, 0.15, 0.05, 0.05, 0.05, 0.05, 0.15]$, which is based on a rule of thumb. If text error correction is applied to documents identified by OCR technology, the weight of shape similarity should be greater. Similarly, if it is applied to documents input by Pinyin Input Method, the weight of sound similarity should be greater.

Analogously, for two Chinese strings a and b of identical length, we calculate the SSCS for characters corresponding to the position. Then the SSCS of two strings is computed as:

$$SSCS(a, b) = \frac{1}{|a|} \sum_{e} SSCS(e, e'). \tag{6}$$

where e and e' are characters in the corresponding positions in the strings a and b.

Table 5 shows an example of four encoded Chinese characters and their corresponding SSCS. Among them, the SSCs of "许" and "徐" are similarly encoded in the sound part, while the SSCs of "轶(yi)" and "秩(zhi)" are similar in the shape part. They both have an SSCS of 0.6, which avoids the drawback that Cosine similarity and Dice similarity simply assign the similarity of different characters as 0. Based on this, we can get the SSCS of two Chinese strings $a =$ "许艺" and $b =$ "徐艺":

$$SSCS(a, b) = (0.6 + 1)/2 = 0.8$$

However, the above proposed SSCS cannot handle the similarity of strings with different lengths, whereas Cosine similarity or Dice similarity can partially improve this problem. Therefore, the main idea of our Hybrid similarity is to consider both the information contained in Dice similarity and SSCS, which lead to better discrimination of field similarity. The Hybrid similarity based on SSCS and Dice coefficient is calculated as:

$$Hybrid(a, b) = \alpha SSCS(a, b) + (1 - \alpha) Dice(a, b). \tag{7}$$

where α is the confidence (weight) of the SSCS, and the value of α varies with different occasions. In this study, we use the following guidelines:

$$\alpha = \begin{cases} 1 & |a| = |b|,\ SSCS(a, b) \geq \tau \\ \delta & |a| = |b|,\ SSCS(a, b) \leq \tau \\ 0 & otherwise \end{cases}$$

where the threshold τ and the weight δ are the hyperparameters. Based on this, for two strings a and b of identical length, we use their SSCS as similarity when their SSCS is greater than or equal to τ, and conversely, we use a weighted similarity of $\alpha = \delta$. In the case of length discrepancies, we apply the Dice similarity directly to calculate the similarity.

As for the earlier example, if let $\tau = 0.8$, the Hybrid similarity between $a =$ "许艺" and $b =$ "徐艺" is computed as:

$$Hybrid(a, b) = SSCS(a, b) = 0.8$$

while their Cosine similarity and Dice similarity are both 0.5, meaning that our Hybrid similarity is obviously more reasonable.

IV. EXPERIMENTS

A. SYNTHETIC DATASETS

To conduct this study, we use synthetic datasets to know the truth of matches -considered as the "gold standard"- against which to assess our linkage decisions. In addition, knowing and controlling data quality (the proportion and type of errors in each data set) is helpful to evaluate the performance of record linkage methods. In practice, such information (the truth of matches, type, and rate of errors) is very difficult to obtain because of the costly verifications required. As shown in Fig.2, we randomly generate two datasets containing real-world noise by the following approaches [23]:

Step 1: We begin by generating a sample with N_E fictitious records. Each of these records consists of four fields: name, address, sex, and a unique identification key (used to determine whether a record pair corresponds to the same entity or not). Here is an example of one record in the sample:

< Name> 张伟 <Name >
< Address> 湖北省武汉市洪山区 < Address>
< Sex> M < Sex>
< ID> 0001 < ID>

Step 2: From these N_E generated records, we constitute the datasets A and B by randomly sampling (without replacement) N_A and N_B records such that $N_A + N_B = \alpha N_E$ with $1 < \alpha < N_E$. These two datasets have therefore N_C common records, with N_C ranging from $(\alpha - 1) \times N_E$ to $min(N_A, N_B)$. In this study, we let $N_E = 1500$ and $N_A = N_B = 1000$.

Step 2: Errors are introduced into a proportion of randomly selected records in datasets A and B (no error is introduced in the identification key). The types of errors introduced in the synthetic datasets include omission, substitution, or transposition of one or more characters among a string field.

These types of errors are chosen because they are the most common spelling errors in identifier fields according to a data validation study [5]. In a data set, these errors will be present in a certain proportion of records which can be up to 36.5% according to the literature [5]. Both the type and proportion of errors can be modulated to construct data sets that could be encountered in real linkage tasks. In addition, we have added some extra errors for Chinese characters as follows:

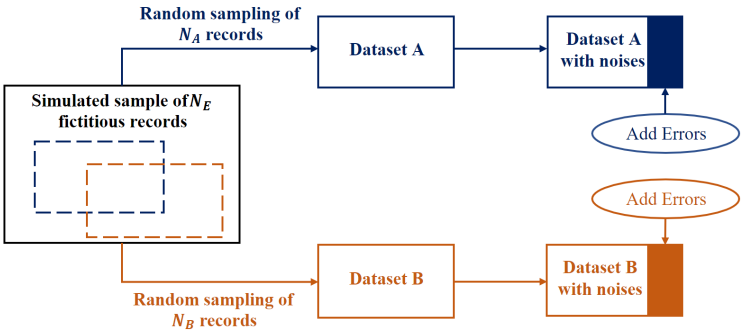


FIGURE 2. The process of generating simulated datasets.

1) SUBSTITUTION

When the information is collected by scanning handwritten characters or verbal inquiries, the error of substitution sometimes occurs. Therefore, we have consulted two Chinese character dictionaries. One is a set of characters with similar pronunciation while the other is a set of characters with similar forms. We then randomly replace characters in the field using these two dictionaries. For example, the record given in Step 1 can be modified in name as ("张" is replaced by "章" because they have the same pronunciation):

< Name> 章伟 <Name >
< Address> 湖北省武汉市洪山区 < Address>
< Sex> M < Sex>
< ID> 0001 < ID>

2) DENORMALIZATION

Information in real data sets can be heterogeneous, especially for fields like addresses. Since different organizations have their own requirements for data quality, the data collected may be incomplete in some parts of a field. Hence, we generate some incomplete information in our simulated datasets. For example, we randomly delete some information in the address field (such as province, city, or district). One example of denormalization in the address is like (province information "湖北省" is omitted):

< Name> 张伟 <Name >
< Address> 武汉市洪山区 < Address>
< Sex> M < Sex>
< ID> 0001 < ID>

B. EXPERIMENTAL DESIGN

We evaluate the effectiveness of our proposed techniques on synthetic datasets. 100 paired datasets are generated by randomly adding errors in the approaches described above. Since the field values of name and address in the data are composed of Chinese characters, we compute their similarity based on the four similarity functions introduced in Section 2 and [21], respectively. One thing to note is that SSC-based methods are only applied when calculating the similarity of names, as the address field is rarely misspelled. With each paired dataset, we then perform record linkage process by using the classification method of PRL-W.

We implement the linkage methods with Python 3.7 and conduct experiments using a computer with a CPU Intel (R) Core (TM) i5 10210U 2.11 GHz and 16 GB RAM.

We start with a preliminary evaluation of our improvement methodology to search the best hyperparameters τ and δ by using the measures of precision, recall, F1-score. An error analysis is then done to demonstrate the effectiveness of the new method. Finally, we make a simple comparison in terms of runtime.

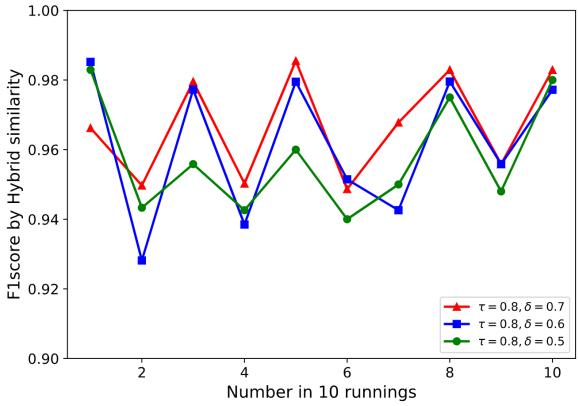


FIGURE 3. F1-score of classification by using different values of δ.

C. RESULTS

In Fig. 3, we present the F1-score of classification based on Hybrid similarity by using different values of δ and using a fixed threshold τ of 0.8, which is the optimal threshold obtained by multiple experiments. At this point, an intuitive interpretation of Hybrid similarity is that we fully trust the SSCS only if the SSCS between strings is greater than or equal to 0.8; otherwise, we partially trust the SSCS. It can be observed that when the weight $\delta = 0.7$, the F1-score of the classification results is the best. In the experiment, we have also tried weights of 0.9 and 1.0, which lead to a significant decrease in F1-score due to the SSCS tend to bring more false positives. Therefore, the Hybrid similarity

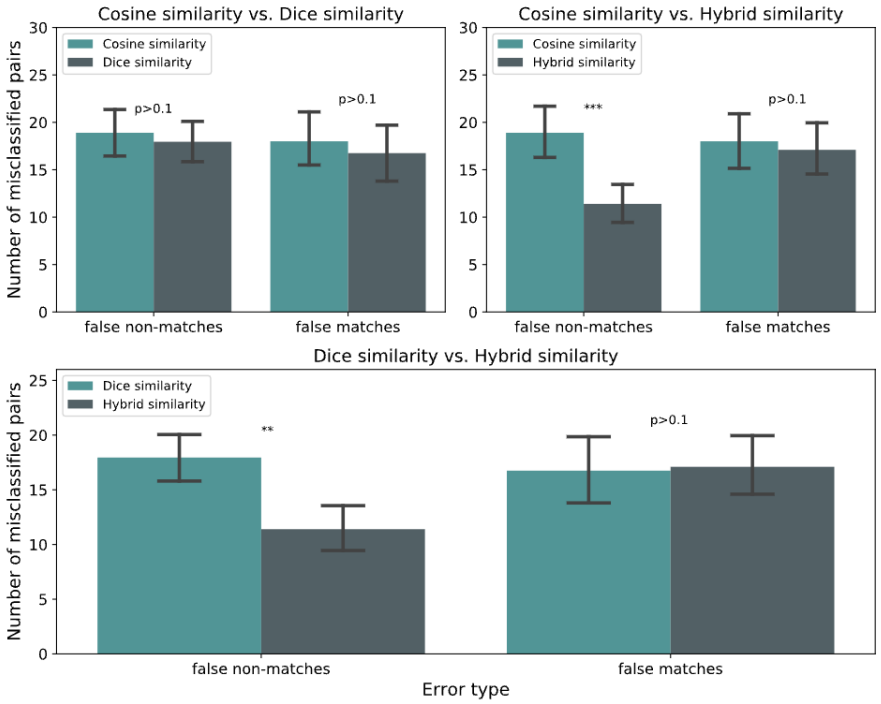


FIGURE 4. H-test for numbers of false matches and false non-matches generated by three methods. We use the following symbols to represent the p-value: 0-0.001: ***, 0.001-0.01: **, 0.01-0.1: *, and when the p-value is greater than 1, we labeled it as p > 1. For example, in the second bar plot, the p-value of Cosine similarity and Hybrid similarity with regard to false non-matches is *** (significant), that is, p < 0.001.

mentioned in this section uses the optimal parameters of $\tau = 0.8, \delta = 0.7$.

To evaluate our method in a scenario where the records are composed of Chinese characters with some errors, in Table 6 we show the performance of previously proposed methods in precision, recall, and F1-score. In addition, we compare the effects of using four-corner-coding (#FC) and WuBi-coding (#WB) separately as part of the SSC. As can be seen, Dice similarity and the method in the previous research [21] (Improved Editing Distance, which referred to as IED in this article) with WuBi-coding achieve the highest precision and recall respectively. However, F1-score is the harmonious value of precision and recall, which is frequently applied to evaluate the effectiveness of the binary classification, and our proposed method leverages both the advantages of Dice similarity and IED, hence performs best in F1-score.

With regard to the effect of four-corner-coding and WuBi-coding in IED and Hybrid similarity, we find that the performances of the latter method are almost better than the former in all three metrics. Overall, the Hybrid similarity method that combines information from SSCS and Dice similarity performs best in record linkage.

To prove that our approach does lead to improvements, an error analysis is performed to examine the ability of the three methods in reducing the number of false matches (record pairs that originally referred to different entities were classified as matches) and false non-matches (record pairs that originally referred to the same entity were classified as non-matches). Since the number of misclassifications does not follow the Gaussian distribution, we determine to apply a commonly used non-parametric test of paired Kruskal-Wallis test (H-test) for the results based on PRL-W method.

TABLE 6. Results of Record Linkage on Synthetic Datasets.

Method	Precision	Recall	F1-score
Cosine similarity	97.15	97.17	97.15
Dice similarity	**97.50**	97.32	97.40
IED(#FC)	97.00	97.83	97.41
Hybrid similarity(#FC)	97.28	98.12	97.69
IED (#WB)	97.02	**98.35**	97.68
Hybrid similarity(#WB)	97.40	98.29	97.84

As shown in Fig. 4, no significant difference is observed in the number of false non-matches and the number of false matches between Cosine similarity and Dice similarity. There is a significant reduction in the number of false non-matches by using the Hybrid similarity(#WB) method, which lead to the least number of false non-matches (11.7 on average out of 10^6 record pairs), while the Cosine similarity and Dice similarity have similar performance (19.2 and 18.7 on average, respectively). As for the average numbers of false

TABLE 7. Results of Name Matching on Synthetic Datasets.

Method	Precision	Recall	F1-score
Cosine similarity	81.98	65.25	72.66
Dice similarity	**98.08**	78.45	87.17
IED(#FC)	82.52	91.27	86.67
Hybrid similarity(#FC)	86.64	90.15	88.36
IED(#WB)	85.12	**93.20**	88.98
Hybrid similarity(#WB)	88.37	90.22	89.28

TABLE 8. Computational Time for Three Methods.

Method	Running time
Cosine similarity	58.42 s
Dice similarity	16.25 s
IED (#WB or #FC)	55.40 s
Hybrid similarity (#WB or #FC)	25.64 s

matches, there is no significant difference between the three methods.

In addition, to explain the impact of different measures of similarity regarding the field name, we performed an additional experiment by using only the field name for matching. Dice similarity and IED with WuBi-coding have the best precision and recall performance as shown in Table 7. However, the best F1-score is shown by the Hybrid similarity with WuBi-coding, which is 89.28%. Similarly, the methods using SSC with WuBi-coding produces the higher precision, recall and F1-score than the four-corner-coding. Therefore, Hybrid similarity with WuBi-coding provides a special similarity measure for field name, which can effectively handle noises (as described in the synthetic datasets in this section) in Chinese names, thus significantly improving the results of record linkage.

In Table 8, we compare the computational time of the four methods. It can be seen that the method based on Dice similarity cost the least computation time (16.25 s for 10^6 record pairs). The method based on Cosine similarity has an expensive computation (58.42 s for 10^6 record pairs). As a compromise between the quality and efficiency of linkage results, our improved approach results in a significant improvement in the quality of classification results at an additional computational cost of only 9.39s compared to the original Dice similarity approach.

V. CONCLUSION

The alphabet-based string comparison algorithm cannot naively handle similarity measurement in the Chinese environment. We detailed the application of Cosine similarity and Dice similarity methods in record linkage, and adapted an encoding technique based on the shape and pronunciation of Chinese characters. We then provided a novel computational method of the combination of SSC and Dice similarity for calculating the similarity between Chinese names.

All four methods were experimented on synthetic datasets. In record linkage, when comparing Chinese characters by using our proposed Hybrid similarity method, it shows its outperformance in reducing the number of false non-matches.

In this article, we have only taken into account 3 commonly used fields in record linkage. Our proposed method demonstrates satisfactory performance in reducing false non-matches. However, for false matches, the Hybrid similarity-based approach barely improves, which is part of what we would like to study in the future.

REFERENCES

[1] P. Christen, *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Berlin, Germany: Springer, 2012.

[2] S. J. Grannis, J. M. Overhage, and C. J. McDonald, "Analysis of identifier performance using a deterministic linkage algorithm," in *Proc. AMIA Annu. Symp. AMIA Symp.*, 2002, pp. 305–309.

[3] B. Li, H. Quan, A. Fong, and M. Lu, "Assessing record linkage between health care and vital statistics databases using deterministic methods," *BMC Health Services Res.*, vol. 6, no. 1, p. 48, Apr. 2006, doi: 10.1186/1472-6963-6-48.

[4] S. L. DuVall, R. A. Kerber, and A. Thomas, "Extending the fellegi–sunter probabilistic record linkage method for approximate field comparators," *J. Biomed. Informat.*, vol. 43, no. 1, pp. 24–30, Feb. 2010, doi: 10.1016/j.jbi.2009.08.004.

[5] C. Friedman and R. Sideli, "Tolerating spelling errors during patient validation," *Comput. Biomed. Res.*, vol. 25, no. 5, pp. 486–509, Oct. 1992, doi: 10.1016/0010-4809(92)90005-U.

[6] E. H. Porter and W. E. Winkler. (1997). *Approximate String Comparison and its Effect on an Advanced Record Linkage System*. Accessed: Dec. 11, 2013. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.7347

[7] M. A. Jaro, "Probabilistic linkage of large public health data files," *Statist. Med.*, vol. 14, nos. 5–7, pp. 491–498, Mar. 1995, doi: 10.1002/sim.4780140510.

[8] M. Tromp, A. C. Ravelli, G. J. Bonsel, A. Hasman, and J. B. Reitsma, "Results from simulated data sets: Probabilistic record linkage outperforms deterministic record linkage," *J. Clin. Epidemiol.*, vol. 64, no. 5, pp. 565–572, May 2011, doi: 10.1016/j.jclinepi.2010.05.008.

[9] N. Méray, J. B. Reitsma, A. C. J. Ravelli, and G. J. Bonsel, "Probabilistic record linkage is a valid and transparent tool to combine databases without a patient identification number," *J. Clin. Epidemiol.*, vol. 60, no. 9, pp. 883.e1–883.e11, Sep. 2007, doi: 10.1016/j.jclinepi.2006.11.021.

[10] I. P. Fellegi and A. B. Sunter, "A theory for record linkage," *J. Amer. Stat. Assoc.*, vol. 64, no. 328, pp. 1183–1210, 1969.

[11] V. J. Zhu, M. J. Overhage, J. Egg, S. M. Downs, and S. J. Grannis, "An empiric modification to the probabilistic record linkage algorithm using frequency-based weight scaling," *J. Amer. Med. Inform. Assoc.*, vol. 16, no. 5, pp. 738–745, Sep. 2009, doi: 10.1197/jamia.M3186.

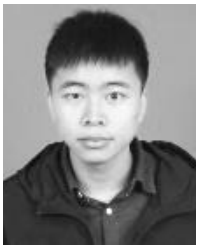
[12] W. E. Winkler. (1990). *String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage*. Accessed: Dec. 6, 2013. [Online]. Available: http://www.eric.ed.gov/ERICWebPortal/recordDetail?accno=ED325505

[13] W. McNaughton, *Reading Writing Chinese*, 3rd ed. Tokyo, Japan: Tuttle Publishing, 2013.

[14] Q. Li, W. An, A. Zhou, and L. Ma, "Recognition of offline handwritten chinese characters using the tesseract open source OCR engine," in *Proc. 8th Int. Conf. Intell. Hum.-Mach. Syst. Cybern. (IHMSC)*, Aug. 2016, pp. 452–456, doi: 10.1109/IHMSC.2016.239.

[15] P. A. Collender, Z. Tom Hu, C. Li, Q. Cheng, X. Li, Y. You, S. Liang, C. Yang, and J. V. Remais, "Machine-learning classifiers for logographic name matching in public health applications: Approaches for incorporating phonetic, visual, and keystroke similarity in large-scale probabilistic record linkage," 2020, *arXiv:2001.01895*. [Online]. Available: http://arxiv.org/abs/2001.01895

[16] C.-L. Liu and J.-H. Lin, "Using structural information for identifying similar chinese characters," in *Proc. 46th Annu. Meeting Assoc. Comput. Linguistics Hum. Lang. Technol. Short Papers HLT*, 2008, pp. 93–96, doi: 10.3115/1557690.1557715.
[17] S. Ro, L. Min, and G. Shi-li, "Similarity calculation of chinese character glyph and its application in computer aided proofreading system," *J. Chin. Comput. Syst.*, vol. 29, no. 10, pp. 1964–1968, 2008.
[18] T.-H. Chang, H.-C. Chen, Y.-H. Tseng, and J.-L. Zheng, "Automatic detection and correction for Chinese misspelled words using phonological and orthographic similarities," in *Proc. 24th Conf. Comput. Linguist. Speech Process. ROCLING*, 2012, pp. 125–139.
[19] M. Liu, V. Rus, Q. Liao, and L. Liu, "Encoding and ranking similar Chinese characters," *J. Inf. Sci. Eng.*, vol. 33, no. 5, pp. 1195–1211, 2017, doi: 10.6688/JISE.2017.33.5.6.
[20] M. Chen, Z. Du, Y. Shao, and H. Long, "Chinese characters similarity comparison algorithm based on phonetic code and shape code," *Inf. Technol.*, vol. 11, pp. 73–75, 2018, doi: 10.13274/j.cnki.hdzj.2018.11.016.
[21] H. Wang, Y. Zhang, L. Yang, and C. Wang, "Chinese text error correction suggestion generation based on SoundShape code," in *Chinese Lexical Semantics*. Cham, Switzerland: Springer, 2020, pp. 423–432, doi: 10.1007/978-3-030-38189-9_44.
[22] V. Thada V. Jaglan, "Comparison of Jaccard, dice, cosine similarity coefficient to find best fitness value for Web retrieved documents using genetic algorithm," *Int. J. Innov. Eng. Technol.*, vol. 2, no. 4, pp. 202–205, 2013.
[23] X. Li, A. Guttmann, S. Cipiere, L. Maigne, J. Demongeot, J.-Y. Boire, and L. Ouchchane, "Implementation of an extended fellegi-sunter probabilistic record linkage method using the jaro-winkler string comparator," in *Proc. IEEE-EMBS Int. Conf. Biomed. Health Informat. (BHI)*, Jun. 2014, pp. 375–379, doi: 10.1109/BHI.2014.6864381.
[24] *Definition of PINYIN*. Accessed: Mar. 8, 2020. [Online]. Available: https://www.merriam-webster.com/dictionary/pinyin
[25] J. Yu, X. Jian, H. Xin, and Y. Song, "Joint embeddings of chinese words, characters, and fine-grained subcharacter components," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2017, pp. 286–291.
[26] *98WuBi*. GitHub. Accessed: Mar. 16, 2020. [Online]. Available: https://github.com/HesperusArcher/98WuBi
[27] *cjkvi/cjkvi-ids*. GitHub. Accessed: Mar. 16, 2020. [Online]. Available: https://github.com/cjkvi/cjkvi-ids



SENLIN XU received the B.S. degree from the Jiangxi University of Finance and Economics, in 2019. He is currently pursuing the M.S. degree in probability and statistics with the College of Science, Huazhong Agricultural University.

MINGFAN ZHENG received the B.S. degree from the Guangdong University of Finance, in 2019. He is currently pursuing the M.S. degree in probability and statistics with the College of Science, Huazhong Agricultural University.

XINRAN LI received the B.S. degree in applied mathematics and the M.S. degree in statistics and data processing from the University of Clermont-Ferrand II, in 2009 and 2011, respectively, and the Ph.D. degree in biostatistics and medical informatics from the University of Clermont-Ferrand I, France, in 2015. He is currently an Associate Professor with the Department of Mathematics and Statistics, Huazhong Agricultural University, China. His research interests include record linkage, text mining, and statistical learning.