Title: Effect of Chinese Characters on Machine Learning for Chinese Author Name Disambiguation: A Counterfactual Evaluation

Authors: Jinseok Kim, Jenna Kim, and Jinmo Kim

1. Jinseok Kim

Institute for Research on Innovation & Science, Survey Research Center, Institute for Social Research
University of Michigan
330 Packard Street, Ann Arbor, MI U.S.A. 48104
jinseokk@umich.edu
ORCID id: 0000-0001-6481-2065

2. Jenna Kim

School of Information Sciences
University of Illinois at Urbana-Champaign
501 E. Daniel Street, Champaign, IL U.S.A. 61820
jkim682@illinois.edu

ORCID id: 0000-0001-7438-448X

3. Jinmo Kim

School of Information Sciences
University of Illinois at Urbana-Champaign
501 E. Daniel Street, Champaign, IL U.S.A. 61820
jinmok2@illinois.edu

[Corresponding Author Information]

Jinseok Kim

Institute for Research on Innovation & Science, Survey Research Center, Institute for Social Research
University of Michigan
330 Packard Street, Ann Arbor, MI U.S.A. 48104-2910
734-763-4994
jinseokk@umich.edu

[Financial Support Information]

This paper was supported by the National Science Foundation (grant number #1917663).

Abstract

Chinese author names are known to be more difficult to disambiguate than other ethnic names because they tend to share surnames and forenames, thus creating many homonyms. In this study, we demonstrate how using Chinese characters can affect machine learning for author name disambiguation. For analysis, 15K author names recorded in Chinese are transliterated into English and simplified by initializing their forenames to create counterfactual scenarios reflecting real-world indexing practices in which Chinese characters are usually unavailable. The results showed that Chinese author names that are highly ambiguous in English or with initialized forenames tend to become less confusing if their Chinese characters are included in the processing. Trained and tested on labeled data with different author name formats, classification algorithms are found to get worse in predicting positive (label match) pairs but better in predicting negative (label nonmatch) pairs as Chinese name spelling becomes English-transliterated and forename-initialized. The positive versus negative pair imbalance caused by increased homonyms seems to influence the behaviors of machine learning algorithms, which enable them to learn disambiguation patterns biased toward dominantly large-sized negative pairs. Our findings indicate that recording Chinese author names in native script can help researchers and digital libraries enhance authority control of Chinese author names that continue to increase in size in bibliographic data.

Keywords: Author name disambiguation; machine learning; Chinese names; authority control

Introduction and Background

Author names in bibliographic data can be ambiguous. More than two authors may have the same name. This is a homonym case. On the other hand, an author may be recorded in two or more name variants. This is a synonym case. Name ambiguity has been a lingering issue in research using bibliographic data for decades since Dr. Garfield pointed out that the tradition in academia of abbreviating author names into their full surname and first forename initial format was causing a lot of confusion about 'Who's Who' in publication records [1]. This author identification problem not only affects the correct counting of publications written by distinct authors but also our understanding of patterns of scientific collaboration, which could lead to flawed evaluation of research production and faulty decisions on science policy [2, 3]. So, distinguishing which names refer to whom in bibliographic data has become a serious task that impact the rigor of academic research and the quality of bibliographic data services that use ambiguous bibliographic data.

To address the name ambiguity in bibliographic data, various methods for disambiguating author names have been tried. Some researchers rely on heuristics such as using full surnames and all forename initials for author identification, which has been a dominant practice in bibliometrics and major bibliographic data services [4, 5]. Meanwhile, others attempt to disambiguate ambiguous names using computational methods such as rule-based programming and machine learning, which have been reported to produce good disambiguation results [6-8]. These efforts have resulted in a plethora of research papers on author name disambiguation and several leading bibliographic data services such as DBLP, Scopus, and Web of Science (WOS) are internally applying author name disambiguation control to provide accurate search results to users.

Although author name disambiguation studies are different in their methods and domains (e.g., author names in computer science, biomedicine, or physics), many of them have pointed out the fact that East Asian – Chinese and Korean - author names are more difficult to disambiguate than other ethnic names because they tend to share surnames and forenames, thus creating many homonym cases [2, 5, 9, 10]. Especially, Chinese author names have attracted special attention from name disambiguation scholars as well as bibliometric researchers because the number of Chinese names in digital libraries keeps growing

as more Chinese scholars engage in research production and leading roles in international collaboration [9, 11, 12]. Many studies have been focused fully or partially on disambiguating Chinese author names by developing disambiguation methods targeted at ambiguous bibliographic data in which homonymous Chinese names are heavily over represented [13, 14]. In addition, a few bibliographic data services such as WOS are parsing Chinese author names with special care into, for example, full surnames and all initials of forename syllables (e.g.., Zhang, Wanhua → Zhang, WH) using customized algorithms, which presumably reduces name ambiguity when Chinese authors are identified by name string matching. These efforts indicate that it is critical to disambiguate Chinese author names accurately to improve both author name disambiguation methods and bibliographic data services.

Regarding Chinese author name disambiguation, a few scholars have suggested that recording Chinese author names in their original characters could reduce name ambiguity [5]. This idea is based on the observation that different Chinese names can have the same English name. For example, Figure 1 illustrates that nine different Chinese names are all transliterated into the same name 'Wang Wei' in English. Such simplification of original Chinese via Romanization has been pointed out as a factor aggravating Chinese name ambiguity [15-17]. This implies that, if Chinese author names are recorded in original script, many of them would become less ambiguous than when transliterated into Romanized names. However, there have been few empirical studies on how much using the native script of Chinese names can reduce name ambiguity in bibliographic data. It is important to answer that question because it can provide not only research insights to improve methods for Chinese author name disambiguation but also a solid ground for bibliographic data services to record Chinese author names in native script for the purpose of enhanced authority control as well as indexing that respects cultural diversity.

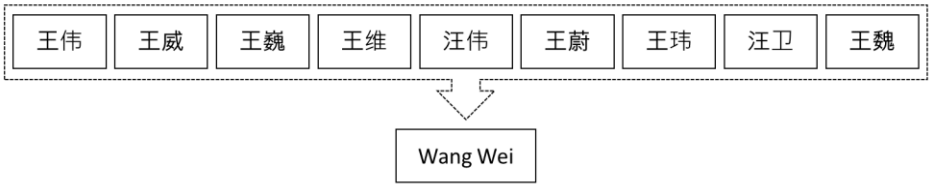

Figure 1. An Illustrated Transliteration of Chinese Author Names into English

As the first attempt for this kind of task, this paper aims to demonstrate how the use of Chinese characters can be helpful in disambiguating Chinese author names. For the purpose, especially, this study takes a counterfactual evaluation approach. In other words, the effectiveness of using Chinese characters in author name disambiguation is proved by measuring how well top-notch disambiguation techniques unable to use Chinese characters (counterfactual) can reduce name ambiguity to the level that is obtainable by using the data in which author names are recorded in Chinese (factual). Specifically, in the example of 'Wang Wei' above, machine learning algorithms are given a task to determine whether instance pairs of 'Wang Wei' refer to the same author or not. Their classification performances are compared with the ones when they are implemented for the same task in which nine Chinese names are available. In addition to using full English names, other two scenarios in which each Romanized Chinese name is simplified into full English surname with first or all forename initial(s) are compared to reflect real world practices in bibliometric research and bibliographic data services. Results show that using Chinese characters in author name disambiguation can be an effective way to substantially improve the performances of machine learning techniques by reducing the number of homonym cases to be disambiguated. In the following section, data collection and machine learning implementation for the counterfactual evaluation are described in detail.

Method

Data

Data Source: A dataset that contains Chinese characters of Chinese author names was collected from the Web of Science Chinese Science Citation Index (CSSI). The CSSI is a database of journals published in the People's Republic of China, which is maintained by the Chinese Academy of Science (CAS) and provided to Clarivate Analytics for international services[i]. For analysis, records of 5,296 research papers (journal articles, reviews, and short papers) published between 2008 and 2017 (10 years) in the field of computer science are retrieved from CSSI. This follows the practice in author name disambiguation research in which computer scientist names have often been the disambiguation target [6, 13].

Data Filtering: Thanks to the curation by Clarivate Analytics in collaboration with CAS, most author names, journal names, and titles in the retrieved dataset are transliterated (author names) or translated (title and journal) into English while conserving the original Chinese script. From the downloaded data, a total of 5,014 paper records are filtered after those without author information are excluded. In the filtered records, a total of 16,849 author name instances are found. Among them, the number of filtered instance after removing the ones that do not contain Chinese characters is 15,554.

Pre-Processing[ii]: Title words are removed if they belong to the list of stop-words[iii], lowercased, and then stemmed[iv]. After lowercased, journal names are cleaned of non-alphabetical characters. Although journal and title information is provided in English by the WOS, Chinese spelling of an author name is transliterated into English by this study's code and divided into comma-delimited surname and forename parts. The process is based on the assumption that the first Chinese character of each name represents a surname, while the remaining characters together a forename. For instance, a Chinese author name, 王晓峰, is transliterated into 'Wang Xiao Feng' and then split into a surname ('Wang') and a forename ('Xiao Feng').

Data Labeling: Many machine learning methods for author name disambiguation require labeled training and test data in which an author name instance is tagged with an author identity (supervised learning). In this study, two human coders manually assign author labels to author name instances. To reduce the amount of labor, first, only instances that share the same Chinese characters are compared, assuming that two instances different in Chinese characters represent different authors. Next, instances with the same Chinese characters are decided by human coders to determine whether it is the same author or not by comparing coauthor names in Chinese, affiliation information, and, if available, online researcher profiles. If two coders disagree on a labeling decision about the same instance, such a conflict is resolved after discussion mediated by the third researcher.

Machine Learning Setups

Disambiguation Scenarios: This study implements disambiguation algorithms on the same data but with four different author name string formats: Chinese characters (Scenario #1; e.g., 王晓峰), full English surname + full English forename (Scenario #2; e.g., Wang, Xiao Feng), full English surname + all forename initials (Scenario #3; e.g., Wang, X F), and full English surname + first forename initial (Scenario #4; e.g., Wang, X). Scenario #1 is a benchmark to compare disambiguation performances of the same algorithms using the same machine learning features but under different name formats. Scenario #2, #3, and #4 are counterfactual (i.e., what if Chinese characters are unavailable?) but also represent real world practices. Most author names in bibliographic data have recently begun to be recorded in full name by publishers [18]. In addition, many disambiguation tasks aim to resolve ambiguity of author names

recorded in English. Thus, Chinese names are transliterated into full English string in Scenario #2. Meanwhile, initializing author forenames in Scenario #3 and Scenario #4 reflects the long-lasting practice of practitioners and researchers using bibliographic data. Specifically, major bibliographic data services like PubMed, SCOPUS, and WOS had not recorded full forenames until mid-2000's or later and provided users with author search results that match on the first or all forename initial(s) plus a full surname [19]. Constrained partially by such practices, bibliometric scholars and academic administrators have relied until now on initialized author names to distinguish author identities [1, 4, 17]. Also, several studies have included initialized names or initialized all forenames in their disambiguation tasks to mimic the aforementioned indexing constraints[20, 21].

Blocking: In most disambiguation studies, only author name instances that meet a certain criterion are compared for disambiguation (blocking). The blocking is used as a preliminary step in disambiguation because it reduces computational burden, while not downgrading much disambiguation performances [8, 9, 22]. Author name instances were put into the same block if they match on Chinese characters (Scenario #1), full English strings (Scenario #2), all forename initials plus full surname (Scenario #3) and the first forename initial plus full surname (Scenario #4). To the best of our knowledge, this paper is the first to use Chinese characters in blocking. Meanwhile, several disambiguation studies have used the full-string-based blocking to disambiguate extremely ambiguous names [14, 23-27]. The first-forename-initial-based blocking is most widely used in disambiguation research. All-forename-initial-based blocking is used because it is a *de facto* disambiguation heuristic in bibliometrics [2, 4] but has rarely been evaluated for its impact as a blocking scheme on machine learning for author disambiguation.

Pairwise Similarity Calculation: This study uses machine learning to predict whether a pair of name instances refer to the same author or not. Such a binary classification approach has been used in many disambiguation studies [6, 21, 28-31]. The labeled list of 15,554 author name instances is randomly split into two subsets with equal sizes – training and test data - for machine learning. Author name instances within the same block are compared pairwisely for their similarity over four features – author name, coauthor names, title, and journal – which are the most commonly used features in disambiguation studies [7, 14, 28, 31].

For Scenario #1, Chinese characters are used for author and coauthor names, while Chinese title and journal names are translated into English (e.g., 计算机科学 → Computer Science). For Scenario #2, #3, and #4, however, author and coauthor names are transliterated English strings, while translated titles and journal names are used like Scenario #1. A feature's text string is first lowercased (except Chinese), ripped of non-alphabetical characters (except Chinese), and dissected into an array of tokens (author and coauthor) or 2~4-grams of alphabetical characters (title and journal). Then, two arrays of tokens or *n*-grams of a pair of instances are calculated for their cosine similarity of token or n-gram frequency. These steps are repeated for both training and test data, producing similarity scores of an instance pair over four features. An example is shown in Table 1. The token or *n*-gram-based segmentation and cosine similarity calculation are used jointly or separately in many disambiguation studies [18, 20, 29, 32-34].

Table 1: A Mock-Up Example of Cosine Similarity Scores for Instance Pairs over Four Features

Block	Pairs	Feature				Label Match?
		Author	Coauthor	Title	Venue	
Wang, Wei	Pair 1	1.00	0.50	0.67	0.12	Match
	Pair 2	1.00	0.25	0.46	0.00	Nonmatch
	Pair 3	0.25	0.00	0.00	0.80	Nonmatch
	Pair *N*	…	….	….	…	Nonmatch

Algorithmic Model Learning: Four classification algorithms – Gradient Boosting, Logistic Regression, Random Forest, and Support Vector Machine - are used for machine learning. These algorithms have been reported to perform best or used as strong baselines in author name disambiguation studies [9, 21, 28, 29, 34-36]. Each algorithm is trained on the list of pairwise similarity scores and labels, as illustrated in Table 1, to learn disambiguation patterns. Based on the learned model, the same algorithm is tested on the list of pairwise similarity scores (without using labels this time) in test data to predict whether a pair of instances refers to the same author (label match) or not (label nonmatch). This learning procedure is implemented using Scikit-learn packages. For Gradient Boosting, 500 estimators are used with max depth=9 and learning rate = 0.125. For Logistic Regression, L2 Regularization with class weight = 1 is chosen. For Random Forest, 500 trees are set after grid search. Linear Kernel is selected for Support Vector Machine. Other hyper-parameter settings for each algorithm can be found at https://scikit-learn.org/stable.

Evaluation

Baseline: Disambiguation results by four classification algorithms are compared with results produced by simple Chinese character matching for each scenario. In other words, two author name instances that have the same Chinese characters are assumed to refer to the same author. As noted above, a few scholars suggested that Chinese characters can be used to distinguish author identities [5, 17].

Performance Measures: The performances of algorithms on test data are evaluated by calculating precision, recall, and F1 for positive (P; label match) and negative (N; label nonmatch) pairs. Precision for positive pairs (Prec-Pos) measures how many predicted positive pairs are correct ones (true positives; TP) over the total number of predicted positive pairs that may contain correct positive pairs (true positives; TP) and incorrect positive pairs (false positives; FP). In contrast, recall for positive pairs (Rec-Pos) measures the ratio of correct positive pairs (true positives; TP) over the total number of true positive pairs that may be predicted correctly as positive (true positives; TP) or incorrectly as negative pairs (false negatives; FN). F1 for positive pairs is a harmonic mean of precision and recall.

$$Precision\ Positive\ (PrecPos) = \frac{Number\ of\ Correctly\ Predicted\ Match}{Number\ of\ Predicted\ Match} = \frac{TP}{(TP + FP)} \quad (1)$$

$$Recall\ Positive\ (RecPos) = \frac{Number\ of\ Correctly\ Predicted\ Match}{Number\ of\ True\ Match} = \frac{TP}{(TP + FN)} \quad (2)$$

$$F1\ Positive = \frac{2 \times RecPos \times PrecPos}{RecPos\ +\ PrecPos} \quad (3)$$

Likewise, precision for negative pairs (Prec-Neg) measures how many predicted negative pairs are correct ones (true negatives; TN) over the total number of predicted negative pairs that may contain correct negative pairs (true negatives; TN) and incorrect negative pairs (false negatives; FN). In contrast, recall for negative pairs (Rec-Neg) measures the ratio of correct negative pairs (true negatives; TN) over the total number of true negative pairs that may be predicted correctly as negative (true negatives; TN) or incorrectly as positive (false positives; FP). F1 for negative pairs is a harmonic mean of precision and recall.

$$Precision\ Negative\ (PrecNeg) = \frac{Number\ of\ Correctly\ Predicted\ Nonmatch}{Number\ of\ Predicted\ Nonmatch} = \frac{TN}{(TN + FN)} \quad (4)$$

$$Recall\ Negative\ (RecNeg) = \frac{Number\ of\ Correctly\ Predicted\ Nonmatch}{Number\ of\ True\ Nonmatch} = \frac{TN}{(TN + FP)} \quad (5)$$

$$F1\ Negative = \frac{2 \times RecNeg \times PrecNeg}{RecNeg\ +\ PrecNeg} \quad (6)$$

Results

Scenario #1

This scenario is a disambiguation task in which all author names are recorded in Chinese. Figure 2 reports the binary classification performances of four algorithms – Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GB), and Support Vector Machine (SVM) – in predicting whether instance pairs are positive (label match) or negative (label nonmatch) pairs. According to Figure 2a, SVM excels other algorithms in precision for predicting positive pairs: its prediction included more accurately predicted positive pairs than others. This is confirmed by comparing the numbers of correctly predicted positive pairs among algorithms reported in Table 2. For example, SVM classified correctly 791 positive pairs (= True Positive; TP) out of 1,098 pairs it predicted as positive (= True Positive + False Positive = TP + FP). Meanwhile, LR classified correctly 1,006 pairs (= TP) out of 1,712 pairs predicted as positive (= TP + FP). Although the number of positive pairs (TP = 1,006) correctly predicted by LR is larger than that by SVM (TP = 791), the ratio of correct positive prediction (TP) over all positive prediction (TP + FP) by SVM (Precision-Positive Score = 0.72) is greater than that by LR (= 0.59), as shown in Figure 2a.

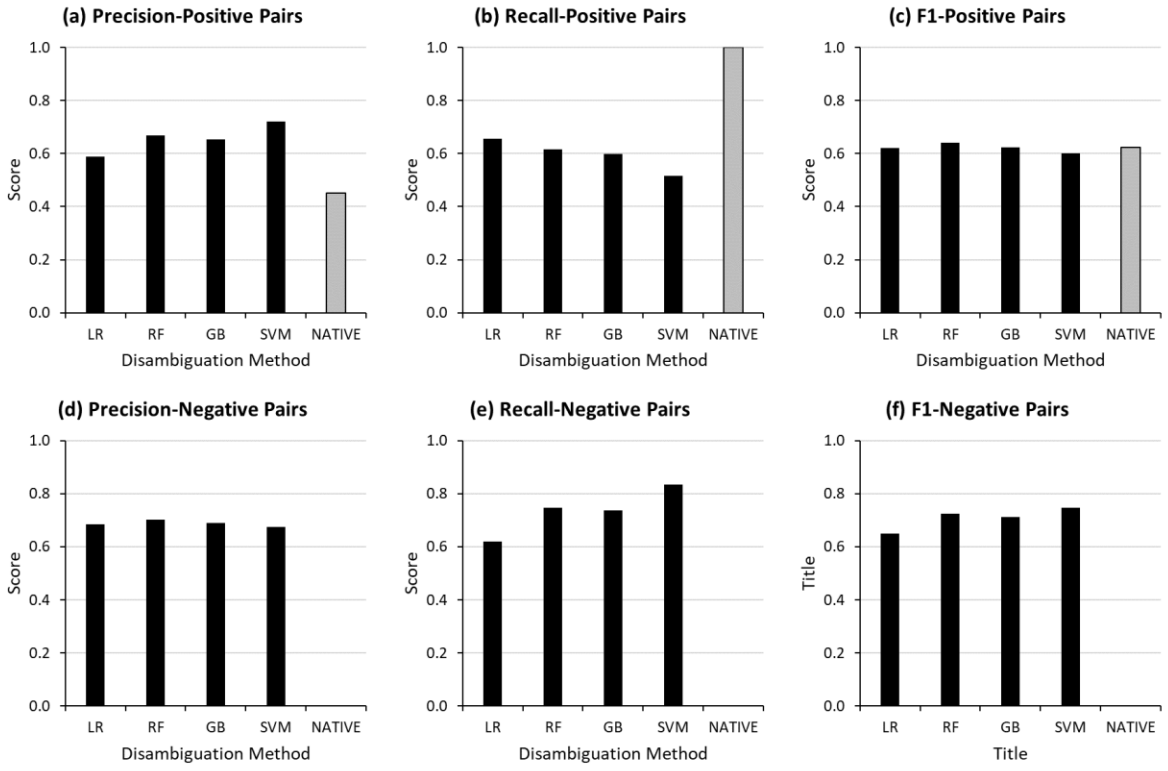

Figure 2. Prediction Performances by Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GB), Support Vector Machine (SVM) and Chinese Character Matching (NATIVE) for Scenario #1

Table 2: Numbers of Correctly or Incorrectly Predicted Pairs by Four Classification Algorithms (LR, RF, GB, and SVM) and Chinese String Matching (Native) for Scenario #1

Disambiguation Method	No. of Pairs	Positive Pairs (1,533)		Negative Pairs (1,857)	
		True Positive (TP)	False Negative (FN)	False Positive (FP)	True Negative (TN)
LR	3,390	1,006	527	706	1,151
RF		946	587	470	1,387
GB		916	617	486	1,371
SVM		791	742	307	1,550
NATIVE		1,533	0	1,857	0

Unlike precision, however, recall in positive pair prediction shows different patterns in Figure 2b. This time, SVM performed worse than others that found more true positive pairs than SVM. In Table 2, for example, SVM classified accurately 791 true positive pairs as positive (TP) and falsely 742 as negative (FN), obtaining a recall score of 0.52 (= Recall-Positive = TP / (TP+FN)). Meanwhile, LR classified successfully 1,006 pairs as positive and falsely 527 as negative, producing a recall of 0.66. This trade-off between precision and recall by each algorithm resulted in similar F1 scores for predicting positive pairs as shown in Figure 2c, in which RF stands as the best algorithm if precision and recall are weighed equally.

It is worth noting the performance comparison between algorithmic and string-based disambiguation methods. In Figure 2b, the positive pair prediction by Chinese string matching (NATIVE) produced the perfect recall (Recall-Positive score = 1.0): the simple heuristic could find all true positive pairs without falsely classifying any of them as negative (FN = 0). This is expected: in Scenario #1, only name instances that share the same Chinese characters are compared because of blocking. So, all name instances within the same block are regarded by the string matching to represent the same author. For this reason, the Chinese string matching could find all true positive pairs (i.e., perfect recall) because true positive pairs in a block all share the same Chinese strings. Thanks to this high recall, NATIVE could produce a similar F1 score to those by four algorithms although its precision was much lower than those by the algorithms.

The string-based matching, however, works against disambiguating name pairs that share the same Chinese string but refer to different authors (Chinese homonyms; negative pairs). It always decides them as positive (FP). Therefore, in Table 2, Chinese string matching (Native) is shown to predict all pairs as positive, whether it be true or false (TP = 1,533 or FP = 1,857), while leaving no false (FN = 0) or true (TN = 0) negatives. That is why no bar was shown for Native in Figure 2d, 2e, and 2f. In contrast, four algorithms showed different precision and recall in predicting negative pairs, although their differences can be slight (precision) or substantial (recall) between the best and worst performers.

Scenario #2

This scenario is a disambiguation task assuming the counterfactual situation in which all author names in training and test data are recorded in English transliterated from Chinese. Figure 3 reports that regarding positive pair prediction, four algorithms and the heuristic showed similar performance patterns that were observed in Figure 2. Some algorithms performed better than others in precision (Figure 3a), which was offset by reduced recall (Figure 3b) leading to not so much differentiated F1 scores. Chinse string matching produced a lower recall than algorithms but a perfect recall, which made its overall performance

similar to those by the algorithms (Figure 3c). Regarding negative pair prediction, also, trade-offs between precision and recall for each algorithm occur, resulting in similar F1 scores across algorithms.

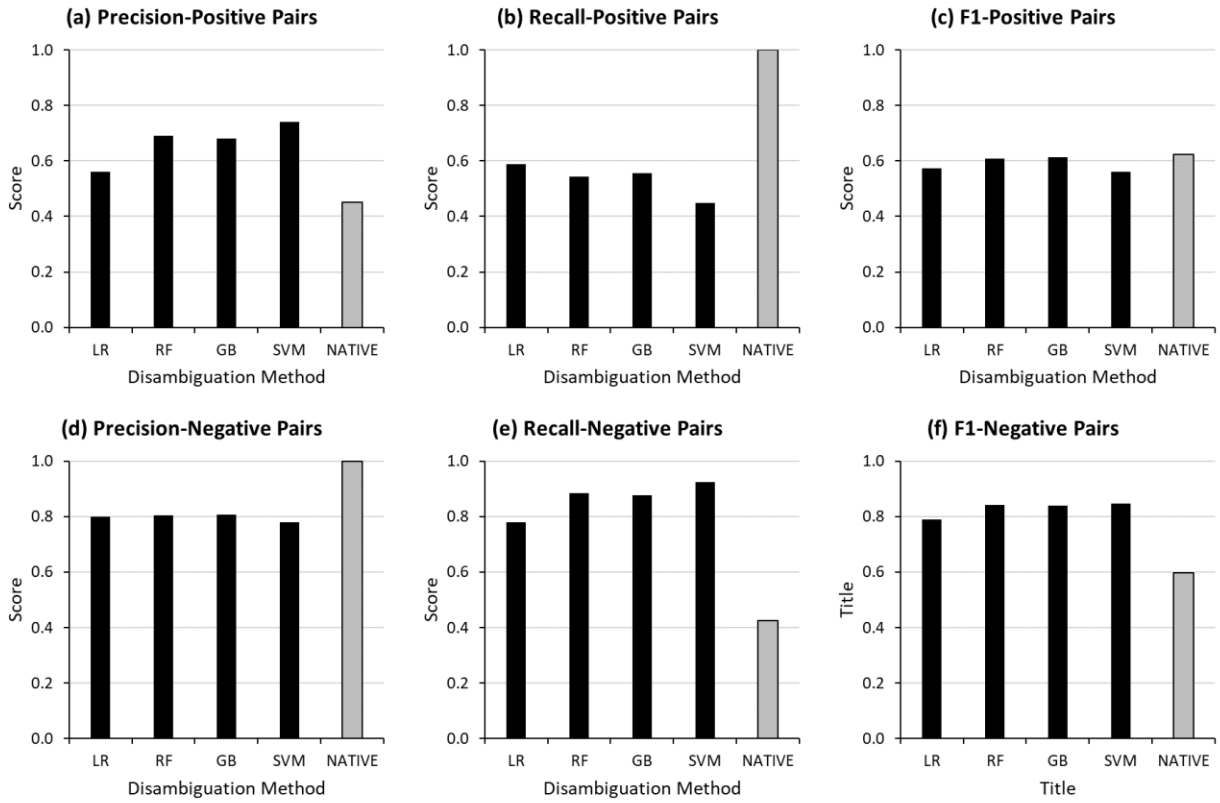

Figure 3: Prediction Performances by Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GB), Support Vector Machine (SVM) and Chinese Character Matching (NATIVE) for Scenario #2

There is a noticeable difference between Scenario #1 and Scenario #2. In Table 3, the number of negative pairs increased from 1,857 in Scenario #1 to 3,236 in Scenario #2 (1,379 newly added), while the number of positive pairs (= 1,533) in both scenarios was unchanged. Moreover, unlike Scenario #1 in which Chinese string match did not produce any negative pair prediction, the baseline method in Scenario #2 was able to predict negative pairs (see gray bars in Figure 3d, 3e, and 3f in contrast to Figure 2d, 2e, and 2f). Figure 4 illustrates the difference by showing how eight author names instances are disambiguated.

Table 3: Numbers of Correctly or Incorrectly Predicted Pairs by Four Classification Algorithms (LR, RF, GB, and SVM) and Chinese String Matching (Native) for Scenario #2

Disambiguation Method	No. of Pairs	Positive Pairs (1,533)		Negative Pairs (3,236)	
		True Positive (TP)	False Negative (FN)	False Positive (FP)	True Negative (TN)
LR	4,769	903	630	708	2,528
RF		834	699	373	2,863
GB		852	681	398	2,838
SVM		689	844	240	2,996
NATIVE		1,533	0	1,857	1,379

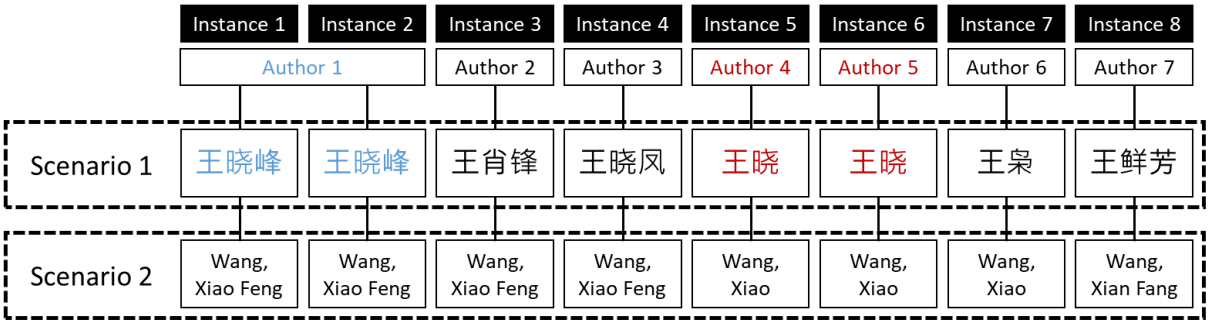

Figure 4. A Mock-Up Example of Disambiguating Author Names in Chinese (Scenario #1) and Transliterated English (Scenario #2)

According to Scenario #1 in Figure 4, instances can have one positive (label match) pair (instance pairs 1-2) and one negative (label nonmatch) pair (instance pairs 5-6). Specifically, two name instances in the '王晓峰' block refer to the same author (Author 1), while those two in the '王晓' block mean different authors (Author 4 & Author 5), respectively. Other instances have no comparable instance when blocked by Chinese characters, meaning that they are not ambiguous. So, the disambiguation task of an algorithm here is to predict whether these two pairs (instance pairs 1-2 and 5-6) refer to the same author (positive) or not (negative). Chinese string matching decides these two pairs as positive because each pair share the same Chinese characters.

According Scenario #2, however, the same set of instances produce one positive pair (instance pairs: 1-2) but eight negative pairs (instance pairs 1-3, 1-4, 2-3, 2-4, and 3-4 for 'Wang, Xiao Feng' block; 5-6, 5-7, and 6-7 for 'Wang, Xiao' block). So, the disambiguation task in Scenario #2 becomes more challenging than Scenario #1's because an algorithm needs to disambiguate eight pairs. The newly added negative pairs are created due to English homonyms ('Wang, Xiao Feng' and 'Wang, Xiao) transliterated from different Chinese names that refer to different authors). In reverse, when Chinese characters are used in disambiguation, homonymous English names can become unambiguous. In this way, Chinese string matching could classify correctly the new 1,379 negative pairs (TN = 1,379) as shown in Table 3 and, combined with its false positive prediction (FP = 1,857), could produce precision (=TN / (TN+FN)), recall (=TN / (TN+FP)), and F1 scores in negative pair prediction.

Scenario #3 & Scenario #4

These two scenarios represent counterfactual situations in which Chinese author names are recorded in simplified English names: a full surname followed by all forename initials (Scenario #3) or the first forename initial (Scenario #4). Figure 5 ~ 6 report the disambiguation performances measured by precision and recall, while Table 4 ~ 5 provide details of prediction results using the confusion matrix dimensions ('true-false-positive-negative'). Like the case of Scenario #2 compared with Scenario #1, the numbers of negative pairs in both scenarios increased substantially: 1,857 (Scenario #1) → 3,236 (Scenario #2) → 14,247 (Scenario #3) → 93,784 (Scenario #4). Figure 7 demonstrates how this dramatic increase can occur for Chinese names.

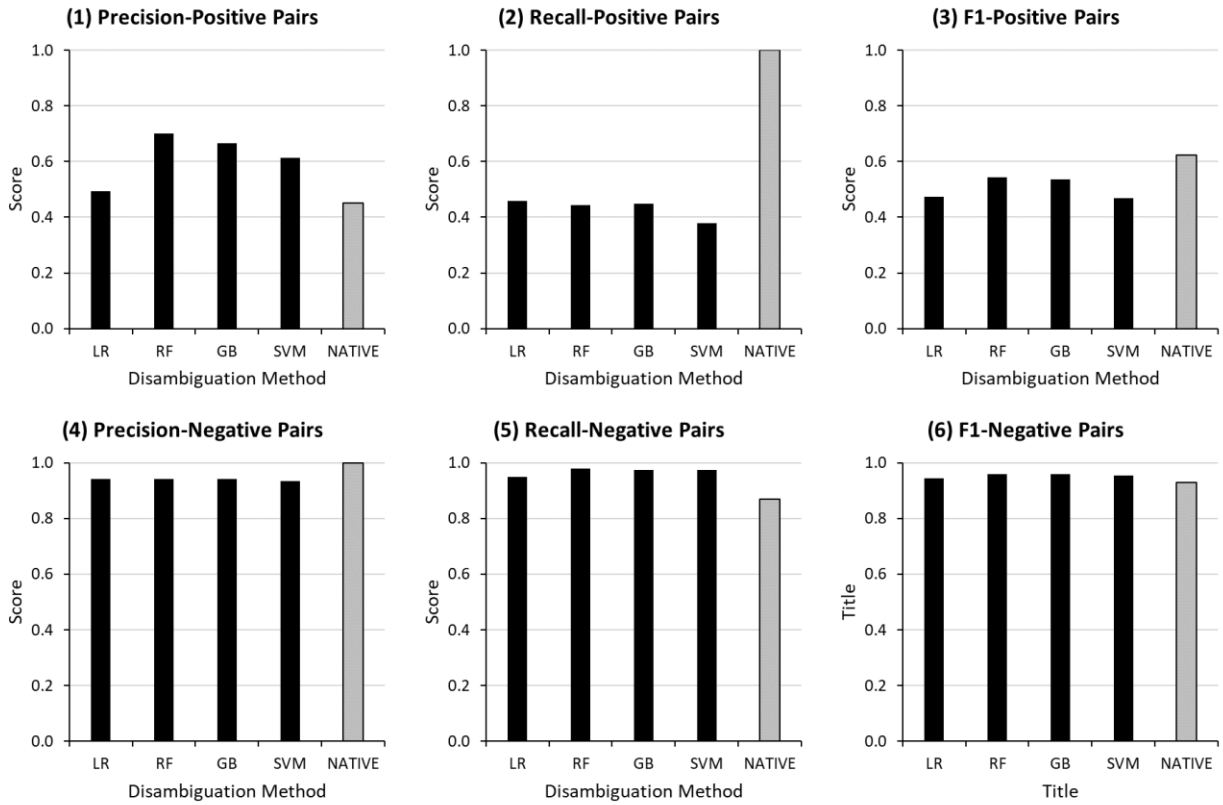

Figure 5. Prediction Performances by Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GB), Support Vector Machine (SVM) and Chinese Character Matching (NATIVE) for Scenario #3

Table 4: Numbers of Correctly or Incorrectly Predicted Pairs by Four Classification Algorithms (LR, RF, GB, and SVM) and Chinese String Matching (Native) for Scenario #3

Disambiguation Method	No. of Pairs	Positive Pairs (1,533)		Negative Pairs (14,247)	
		True Positive (TP)	False Negative (FN)	False Positive (FP)	True Negative (TN)
LR	15,780	702	831	723	13,524
RF		679	854	290	13,957
GB		689	844	347	13,900
SVM		581	952	367	13,880
NATIVE		1,533	0	1,857	12,390

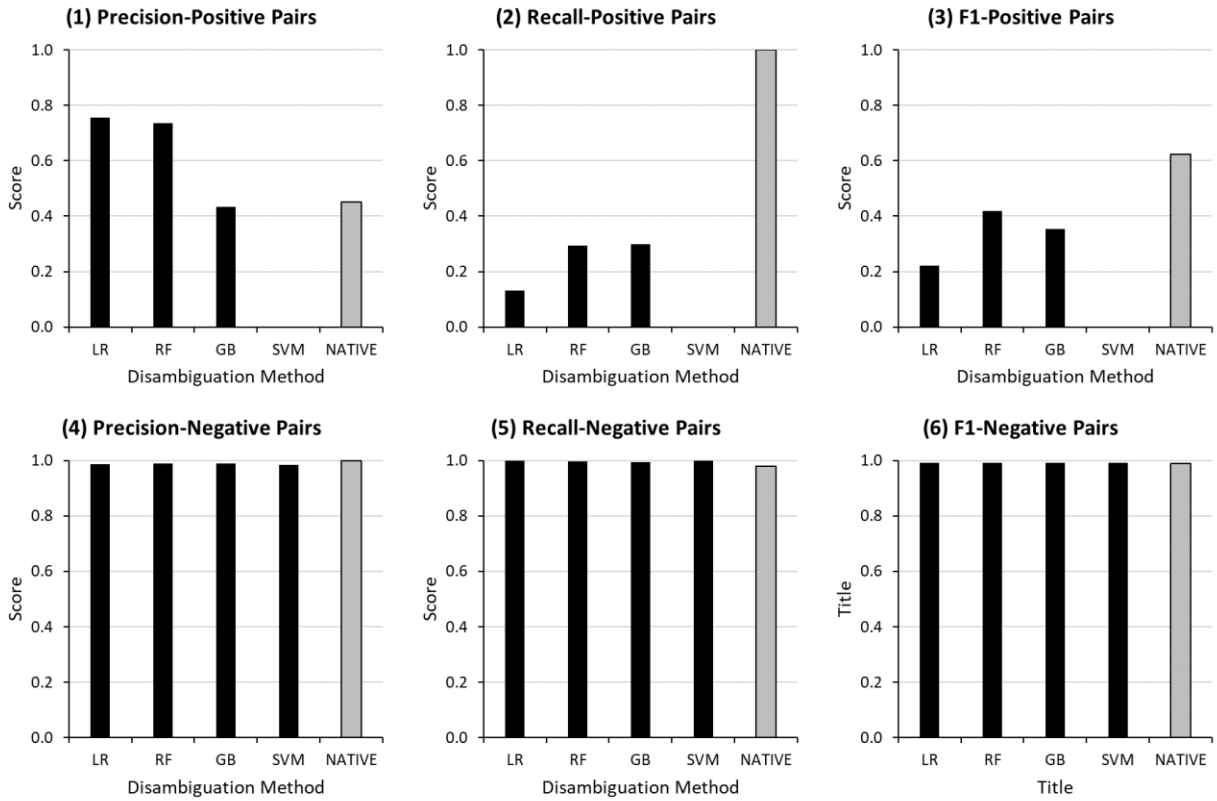

Figure 6. Prediction Performances by Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GB), Support Vector Machine (SVM) and Chinese Character Matching (NATIVE) for Scenario #4

Table 5: Numbers of Correctly or Incorrectly Predicted Pairs by Four Classification Algorithms (LR, RF, GB, and SVM) and Chinese String Matching (Native) for Scenario #4

Disambiguation Method	No. of Pairs	Positive Pairs (1,533)		Negative Pairs (93,784)	
		True Positive (TP)	False Negative (FN)	False Positive (FP)	True Negative (TN)
LR	95,317	200	1,333	65	93,719
RF		449	1,084	161	93,623
GB		459	1,074	602	93,182
SVM		0	1,533	0	93,784
NATIVE		1,533	0	1,857	91,927

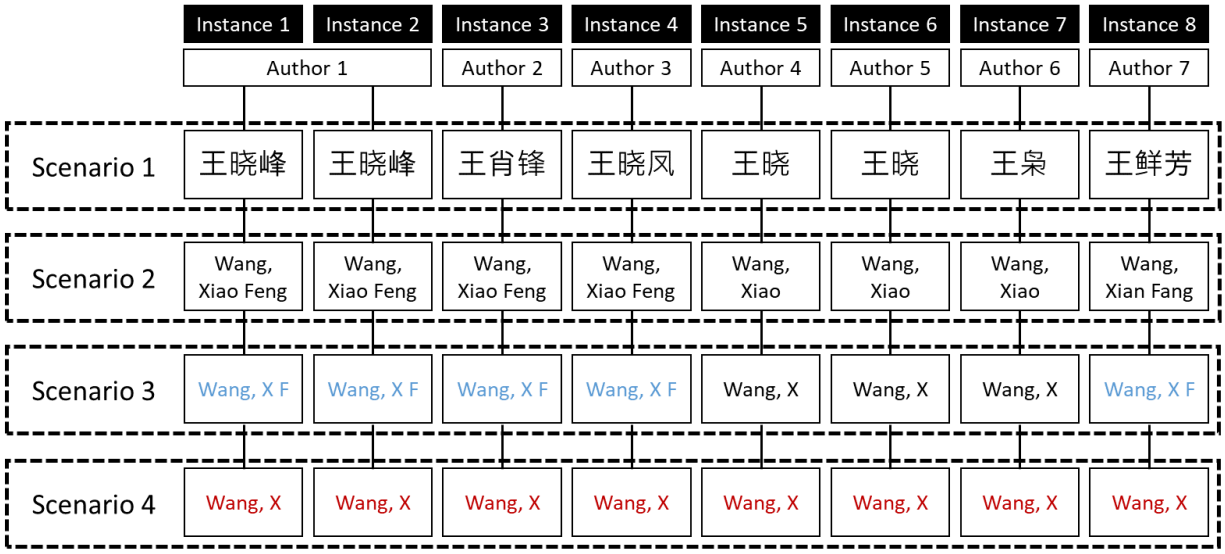

Figure 7. A Mock-Up Example of Disambiguating Author Names in Chinese (Scenario #1), Transliterated English (Scenario #2), Surname Plus All Forename Initials (Scenario #3), and Surname Plus First Forename Initial (Scenario #4)

Figure 7 extends the mock-up example of Figure 4 by adding the cases of Scenario #3 and Scenario #4. By initializing all available forenames (Scenario #3), Instance 1 ~ 4 and Instance 8 are compared together within the 'Wang, X F' block, producing one positive pair (instance pair 1-2) and nine negative pairs[v]. In addition, Instance 5 ~ 7 are compared within the 'Wang, X' block, generating 3 negative pairs[vi]. When the comparable pairs from two blocks are added up, the total is 13 pairs (= 1 positive + 12 negatives). Meanwhile, the fourth scenario converts all instances into 'Wang, X' and make them all to be compared within a single block. As a result of pairwise comparison within the 'Wang, X' block, one positive pair and 27 negative pairs[vii] are generated to be disambiguated (total: 28). To sum up, starting from one in Scenario #1, the numbers of negative pairs in Figure 7 increase to 8 (Scenario #2), 12 (Scenario #3), and 27 (Scenario #4).

The example in Figure 7 illustrates that Chinese names can become more and more ambiguous depending on how name strings are pre-processed. Once transliterated into English, many Chinese author names that are distinguishable by native characters become homonyms that necessitates disambiguation. As names are simplified by forename initialization, more Chinese names become homonyms, aggravating name ambiguity among them especially by increasing negative pairs to be disambiguated. Then, how does the change of negative pair size affect performances of machine learning in predicting positive and negative pairs? Figure 8 is the combination of four figures reported above for each scenario - Figure 2 (Scenario #1), Figure 3 (Scenario #2), Figure 5 (Scenario #3), and Figure 6 (Scenario #4) - to compare the prediction performances by four classification algorithms and Chinese character matching across four scenarios.

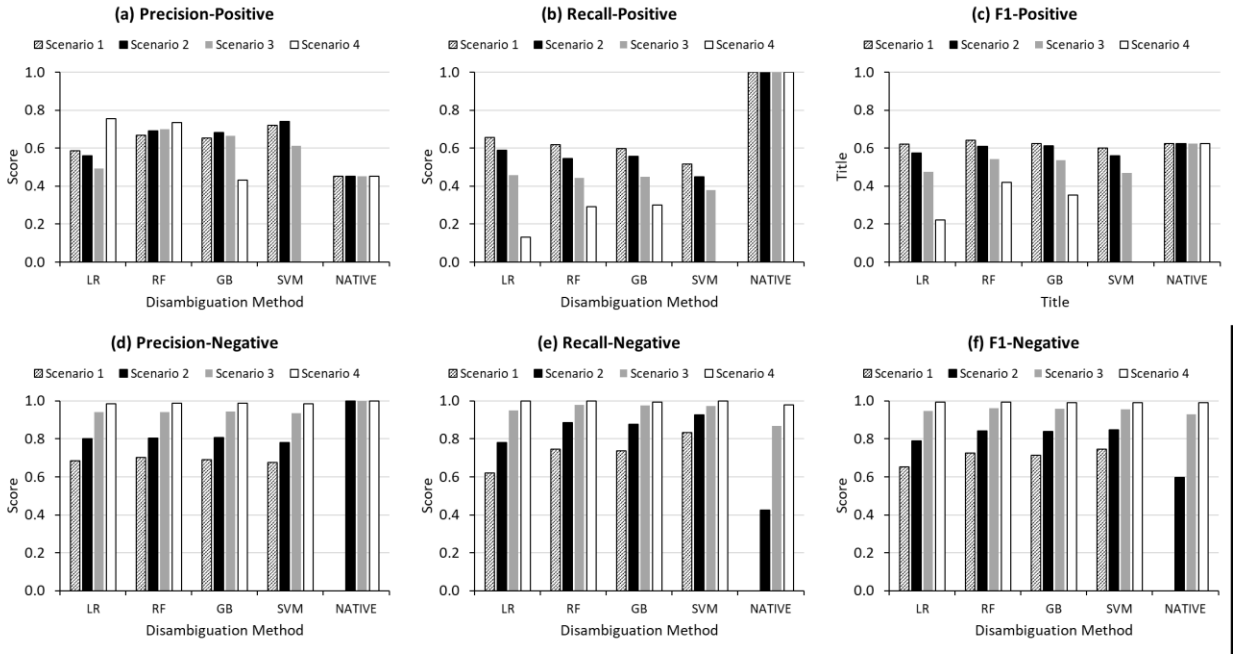

Figure 8. Prediction Performances by Logistic Regression (LR), Random Forest (RF), Gradient Boosting (GB), Support Vector Machine (SVM) and Chinese Character Matching (NATIVE) Compared Across Four Scenarios

According to Figure 8a, four algorithms showed a mixture of rise and fall in precision for predicting positive pairs depending on scenarios. In contrast, Figure 8b showed that their recall in positive pair prediction consistently went downward from Scenario #1 to #4. The diminishing recalls came from the combination of decreased true positives and increased false negatives as shown in Table 2 ~ 5. This observation indicates that machine learning models became less effective in predicting positive pairs. Considering that the number of to-be-disambiguated positive pairs are the same across scenarios, the increased negative pairs are speculated to adversely affect machine learning performances. In other words, algorithmic models seem to be biased toward negative pair patterns due to the heavy imbalance of positive (label-match) versus negative (label-nonmatch) pairs during training and test procedures.

Algorithmic disambiguation performances in negative pair prediction indicates that such a speculation is plausible. As the number of negative pairs increased over scenarios, algorithms produced similar number of false positives or even reduced them, while predicted correctly most negative pairs as negative (TN) despite their increased sizes. In Table 2, for example, LR produced 706 false positives when it was tested to predict 1,857 negative pairs but, in Table 5, only 65 false positives among 93,784 negative pairs. This means that LR learned quite well patterns of negative pairs and predicted also well accordingly, when it is assumed that feature distributions for negative and positive pairs in both training and test data are similar. In contrast, algorithms that performed well in negative pair prediction did not do so much in positive pair prediction. As the number of negative pairs increased, algorithms came to falsely predict more and more positive pairs as negative (FN). For example, LR produced 527 false negatives in Scenario #1 (Table 2), 630 in Scenario #2 (Table 3), 831 in Scenario #3 (Table 4), and 1,333 in Scenario #4 (Table5). This implies that, with increased negative pairs, LR came to decide instance pairs more likely to be negative. Especially, such an imbalance of positive and negative pairs seemed to cause SVM to predict all pairs as negative (TP = 0; FP = 0), which explains why its precision and recall in positive pair prediction did not appear at all in Figure 8a, 8b, and 8c.

These observations indicate that Chinese characters can improve performances of machine learning for Chinese author name disambiguation. They can reduce the number of ambiguous pairs to be disambiguated by reducing negative pairs between homonyms that are created through transliterated English names and initialized forenames. In addition, Chinese characters can also serve as a simple heuristic to effectively distinguish author names. Chinse string matching determines whether all instances sharing the same Chinese characters represent the same authors or not. In Figure 8, this resulted in high recall (because every positive pairs share the same names) but mediocre precision (because sharing the same Chinese characters is not always indicative of the same author identity → Chinese homonyms) in positive pair prediction. As the numbers of positive pairs are the same across scenarios, the precision, recall, and F1 scores by this heuristic are the same across scenarios (i.e., the same bar heights by NATIVE in each Figure 8a, 8b, and 8c). Regarding predicting negative pairs, however, Chinese character matching miss-classified some negative pairs sharing the same Chinese characters as positive (FP). But its accuracy in detecting true negatives became higher (Figure 8d) because most of the increased negative pairs by transliteration (Scenario #2) and forename initialization (Scenario #3 & #4) shared no Chinese characters and, thus, could be easily decided as negative pairs by the Chinese string matching. Depending on the size of newly added negative pairs, the recall in negative pair prediction by Chinese string matching (= TN / (TN + FP)) showed different scores (the larger the negative pair size, the bigger recall score due to the increased TN in the denominator part) in Figure 8e while the false positive size (FP = 1,857) is constant.

Conclusion and Discussion

This study showed how using Chinese characters can affect disambiguation results by algorithms trained and tested on labeled data in which Chinese author names are recorded in Chinese. To illustrate how effective Chinese characters can be in improving disambiguation results, the same names in labeled data were transliterated into full English names and then their forenames were initialized to create counterfactual scenarios in which Chinese characters were unavailable. As homonyms increased in size through transliteration and forename initialization, four classification algorithms produced worse performances in predicting positive (label match) pairs, while got better at predicting negative (label nonmatch) pairs. This diminishing performance in positive pair prediction was considered to arise from the increased negative pairs that affect algorithms to learn models biased toward per-feature similarity patterns that are prominent in negative pairs. Matching Chinese characters was also shown to be an effective heuristic to decide whether a pair of name instances refer to the same author or not, although it could achieve high recall at the cost of low precision in predicting positive pairs or vice versa in predicting negative pairs. Such decent performance by Chinese character matching was possible because name instances that were extremely ambiguous when recorded in full English name or with initialized forenames became less confusing if their Chinese strings were available.

The results reported in this paper can provide several implications for improving author name disambiguation research. First, the problem that the forename initialization of author names can increase name ambiguity greatly has recently been discussed in several studies [2, 20, 22, 37]. This study uniquely investigated the problem in the context of Chinese author names which have been reported to constitute the majority of ambiguous names in bibliographic data when forenames are initialized [2, 5, 9]. Even if recorded in full names, Chinese author names are more ambiguous than other ethnic names, which has led to the over-representation of Chinese author names in labeled data for many studies that aim to disambiguate challenging names [23, 24, 27]. This study helps us to understand better their findings and motivations by illustrating how the name ambiguity of Chinese author names can aggravate depending on the different indexing choices – native Chinese string, transliterated English string, and forename initialization – which were shown to produce different numbers of homonyms that make the same

disambiguation task challenging at different levels. Second, this research also confirms that using name string recorded in more complete format can improve disambiguation performances substantially or make a disambiguation task easier [13, 18, 38]. Taking one step further, this study demonstrated that using Chinese characters (Scenario #1) can boost performances for disambiguating Chinese author names that are highly ambiguous even if their full names are recorded in English (Scenario #2). Third, this study suggests that practitioners as well as disambiguation researchers can use the distinctiveness of Chinese characters to enhance authority control in bibliographic data. A few digital libraries such as American Physical Society (Physical Reviews) and the American Mathematical Society (MathSciNet) allow users to record names in native script, respecting diverse naming cultures. At least for Chinese authors, this practice can help digital libraries to distinguish better Chinese author names and thus produce more accurate results for author name based queries by users. Interestingly, most Korean author names, which are often regarded as ambiguous as Chinese names, can be written in two different ways: one in Korean and the other in Chinese characters (which are different from modernized Chinese characters used by Chinese authors). This implies that using native strings can be quite beneficial to disambiguating Chinese and Korean author names.

To realize such potentials of using Chinese characters in author name disambiguation, however, several issues need to be addressed. First, the findings in this paper were based on scenarios in which all author name instances in each scenario are in the same format. But this may not be the case in digital libraries. Especially, in Scenario #1, Chinese characters were available for all author names. In-depth studies are required to understand how Chinese characters contribute to disambiguation tasks in which varying proportions of name instances are recorded in Chinese. Second, for a counterfactual evaluation, all name instances were converted into simple formats starting from Chinese characters to (transliterated) full English string to forename-initialized string. This pre-processing cannot reflect synonyms due to Chinese spelling errors, transliterated name variants, or flipped name order in real world bibliographic data (e.g., 'Wang, Wei' vs 'Wei, Wang'). How much frequent and consequential such synonym cases are in disambiguating Chinese author names needs to be considered when disambiguation methods for Chinese author names are developed. Finally, the list of 15,554 name instances used in this research may be enough to demonstrate, as an exploratory analysis, the effectiveness of Chinese characters in author disambiguation under controlled settings. Considering the scale and increasing pace of Chinese author names entering digital libraries, homonyms in Chinese may be quite prevalent in large-scale bibliographic data, which could diminish the distinctive power of Chinese characters. Future studies on these issues will help guide researchers and practitioners who develop methods for Chinese author name disambiguation that utilize Chinese characters. In addition, this study can motive scholars to study more on distinguishing author names from other regions than China and Korea using native characters and morphological features[13, 39], which will enrich disambiguation methods enabling us to address name ambiguity originated from diverse naming cultures around the world.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

References

1. Garfield E. British quest for uniqueness versus American egocentrism. *Nature* 1969; 223: 763. DOI: 10.1038/223763b0.
2. Kim J and Diesner J. Distortive effects of initial-based name disambiguation on measurements of large-scale coauthorship networks. *J Assoc Inf Sci Tech* 2016; 67: 1446-1461. DOI: 10.1002/asi.23489.
3. Harzing AW. Health warning: might contain multiple personalities-the problem of homonyms in Thomson Reuters Essential Science Indicators. *Scientometrics* 2015; 105: 2259-2270. DOI: 10.1007/s11192-015-1699-y.
4. Milojević S. Accuracy of simple, initials-based methods for author name disambiguation. *Journal of Informetrics* 2013; 7: 767-773. DOI: 10.1016/j.joi.2013.06.006.
5. Strotmann A and Zhao DZ. Author name disambiguation: What difference does it make in author-based citation analysis? *J Am Soc Inf Sci Tec* 2012; 63: 1820-1833. DOI: 10.1002/asi.22695.
6. Ferreira AA, Gonçalves MA and Laender AHF. A Brief Survey of Automatic Methods for Author Name Disambiguation. *Sigmod Rec* 2012; 41: 15-26.
7. Schulz J. Using Monte Carlo simulations to assess the impact of author name disambiguation quality on different bibliometric analyses. *Scientometrics* 2016; 107: 1283-1298. DOI: 10.1007/s11192-016-1892-7.
8. Hussain I and Asghar S. A survey of author name disambiguation techniques: 2010–2016. *The Knowledge Engineering Review* 2017; 32: e22. 2017/12/05. DOI: 10.1017/S0269888917000182.
9. Torvik VI and Smalheiser NR. Author name disambiguation in MEDLINE. *Acm T Knowl Discov D* 2009; 3 2010/01/15. DOI: 10.1145/1552303.1552304.
10. Wu J and Ding XH. Author name disambiguation in scientific collaboration and mobility cases. *Scientometrics* 2013; 96: 683-697. DOI: 10.1007/s11192-013-0978-8.
11. Zhang Z, Rollins JE and Lipitakis E. China's emerging centrality in the contemporary international scientific collaboration network. *Scientometrics* 2018. journal article. DOI: 10.1007/s11192-018-2788-5.
12. Yuan L, Hao Y, Li M, et al. Who are the international research collaboration partners for China? A novel data perspective based on NSFC grants. *Scientometrics* 2018; 116: 401-422. journal article. DOI: 10.1007/s11192-018-2753-3.
13. Müller MC, Reitz F and Roy N. Data sets for author name disambiguation: An empirical analysis and a new resource. *Scientometrics* 2017; 111: 1467-1500. 2017/06/10. DOI: 10.1007/s11192-017-2363-5.
14. Hussain I and Asghar S. DISC: Disambiguating homonyms using graph structural clustering. *Journal of Information Science* 2018; 44: 830-847. DOI: 10.1177/0165551518761011.
15. Tang L and Walsh JP. Bibliometric fingerprints: name disambiguation based on approximate structure equivalence of cognitive maps. *Scientometrics* 2010; 84: 763-784. DOI: 10.1007/s11192-010-0196-6.
16. Chin W-S, Juan Y-C, Zhuang Y, et al. Effective string processing and matching for author disambiguation. *Proceedings of the 2013 KDD Cup 2013 Workshop*. Chicago, Illinois: Association for Computing Machinery, 2013, p. Article 7.
17. Qiu J. Scientific publishing: Identity crisis. *Nature* 2008; 451: 766-767. DOI: 10.1038/451766a.
18. Kim J and Kim J. Effect of Forename String on Author Name Disambiguation. *J Assoc Inf Sci Tech* In print. DOI: 10.1002/asi.24298.
19. Smalheiser NR and Torvik VI. Author Name Disambiguation. *Annu Rev Inform Sci* 2009; 43: 287-313.
20. Kim J and Kim J. The impact of imbalanced training data on machine learning for author name disambiguation. *Scientometrics* 2018; 117: 511-526. journal article. DOI: 10.1007/s11192-018-2865-9.

21. Han H, Giles L, Zha H, et al. Two Supervised Learning Approaches for Name Disambiguation in Author Citations. In: *JCDL 2004: Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries* Tucson, Arizona, 2004, pp.296-305.
22. Kim K, Sefid A and Giles CL. Scaling Author Name Disambiguation with CNF Blocking. *arXiv preprint arXiv:170909657* 2017.
23. Momeni F and Mayr P. Evaluating Co-authorship Networks in Author Name Disambiguation for Common Names. In: *20th international Conference on Theory and Practice of Digital Libraries (TPDL 2016)* (eds Fuhr N, Kovacs L, Risse T, et al.), Hannover, Germany, 2016, pp.386-391. Springer.
24. Zhang Y, Zhang F, Yao P, et al. Name Disambiguation in AMiner: Clustering, Maintenance, and Human in the Loop. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. London, United Kingdom: ACM, 2018, p. 1002-1011.
25. Wang X, Tang J, Cheng H, et al. ADANA: Active Name Disambiguation. *2011 IEEE 11th International Conference on Data Mining*. IEEE Computer Society, 2011, p. 794-803.
26. Onodera N, Iwasawa M, Midorikawa N, et al. A Method for Eliminating Articles by Homonymous Authors From the Large Number of Articles Retrieved by Author Search. *J Am Soc Inf Sci Tec* 2011; 62: 677-690. DOI: 10.1002/asi.21491.
27. Ackermann MR and Reitz F. Homonym Detection in Curated Bibliographies: Learning from DBLP's Experience. In: *International Conference on Theory and Practice of Digital Libraries (TPDL) 2018* Porto, Portugal, 2018, pp.59-65. Springer International Publishing.
28. Song M, Kim EHJ and Kim HJ. Exploring Author Name Disambiguation on PubMed-Scale. *Journal of Informetrics* 2015; 9: 924-941. DOI: 10.1016/j.joi.2015.08.004.
29. Treeratpituk P and Giles CL. Disambiguating Authors in Academic Publications using Random Forests. *JCDL 2009: Proceedings of the 2009 Acm/Ieee Joint Conference on Digital Libraries* 2009: 39-48.
30. Vishnyakova D, Rodriguez-Esteban R, Ozol K, et al. Author Name Disambiguation in MEDLINE Based on Journal Descriptors and Semantic Types. In: *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)* Osaka, Japan, dec 2016, pp.134-142. The COLING 2016 Organizing Committee.
31. Sanyal DK, Bhowmick PK and Das PP. A review of author name disambiguation techniques for the PubMed bibliographic database. *Journal of Information Science* 2019. DOI: 10.1177/0165551519888605.
32. Han H, Zha HY and Giles CL. Name disambiguation in author citations using a K-way spectral clustering method. *Proceedings of the 5th ACM/IEEE Joint Conference on Digital Libraries, Proceedings* 2005: 334-343. DOI: Doi 10.1145/1065385.1065462.
33. Levin M, Krawczyk S, Bethard S, et al. Citation-Based Bootstrapping for Large-Scale Author Disambiguation. *J Am Soc Inf Sci Tec* 2012; 63: 1030-1047. DOI: 10.1002/asi.22621.
34. Louppe G, Al-Natsheh HT, Susik M, et al. Ethnicity Sensitive Author Disambiguation Using Semi-supervised Learning. *Comm Com Inf Sc* 2016; 649: 272-287. DOI: 10.1007/978-3-319-45880-9_21.
35. Kim K, Sefid A, Weinberg BA, et al. A Web Service for Author Name Disambiguation in Scholarly Databases. In: *2018 IEEE International Conference on Web Services (ICWS)* 2018, pp.265-273. IEEE.
36. Wang J, Berzins K, Hicks D, et al. A boosted-trees method for name disambiguation. *Scientometrics* 2012; 93: 391-411. journal article. DOI: 10.1007/s11192-012-0681-1.
37. Fegley BD and Torvik VI. Has large-scale named-entity network analysis been resting on a flawed assumption? *PLoS One* 2013; 8. DOI: 10.1371/journal.pone.0070299.
38. Backes T. The Impact of Name-Matching and Blocking on Author Disambiguation. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. Torino, Italy: ACM, 2018, p. 803-812.

39. Tatar S and Cicekli I. Automatic rule learning exploiting morphological features for named entity recognition in Turkish. *Journal of Information Science* 2011; 37: 137-151. Article. DOI: 10.1177/0165551511398573.

[i] https://clarivate.com/webofsciencegroup/solutions/webofscience-chinese-science-citation-index/

[ii] The code used for parsing and pre-processing the downloaded data is publicly available at (TBA) for validation and reuse. Note that the downloaded data are not sharable due to the restrictions of WOS data policies.

[iii] https://github.com/stanfordnlp/CoreNLP/blob/master/data/edu/stanford/nlp/patterns/surface/stopwords.txt

[iv] For stemming, Porter's algorithm was used at https://tartarus.org/martin/PorterStemmer/

[v] Negative instance pairs: 1-3, 1-4, 1-8, 2-3, 2-4, 2-8, 3-4, 3-8, 4-8.

[vi] Negative instance pairs: 5-6, 5-7, 6-7

[vii] Negative instance pairs: 1-3, 1-4, 1-5, 1-6, 1-7, 1-8, 2-3, 2-4, 2-5, 2-6, 2-7, 2-8, 3-4, 3-5, 3-6, 3-7, 3-8, 4-5, 4-6, 4-7, 4-8, 5-6, 5-7, 5-8, 6-7, 6-8, 7-8