Advanced Review

Record linkage

Thomas H. Herzog,[1] Fritz Scheuren[2] and William E. Winkler[3]*

This article describes methods for matching duplicates within or across files using non-unique identifiers such as first name, last name, date of birth, address, and other characteristics. © 2010 John Wiley & Sons, Inc. *WIREs Comp Stat*

Record linkage, in the present context, is simply the bringing together of information from two records that are believed to relate to the same entity—for example, the same individual, the same family, or the same business. This might involve the linking of records within a single database to identify duplicate case records. Alternatively, record linkage might involve the linking of records across two or more databases. Such work might be undertaken to merge these databases into a single database with improved coverage or scope. The record linkage work is easiest when unique identification numbers (such as Social Security Numbers) are readily available. The work is more challenging when only *quasi-identifiers* such as given name, surname, date of birth, and address are available. In combination, quasi-identifiers may uniquely identify an individual.

APPLICATIONS OF RECORD LINKAGE

Record linkage is widely used by both businesses and government agencies. Businesses might use record linkage techniques to remove duplicate entries from mailing lists, thereby reducing both printing and mailing costs, and otherwise operating more efficiently. Businesses might also use record linkage to improve the functionality of their databases. For example, in the corporate combination between Bank of America and Countrywide Mortgage, it was widely conjectured that a major factor driving the deal was Bank of America's desire to obtain e-mail addresses of its retail customers from Countrywide Mortgage's database.

Government agencies are frequently concerned with large-scale sample surveys and censuses. In such work, it is critical to the success of the project that the list frame of each survey has few, if any, duplicate records and that the list frame be complete in the sense that all the entities of interest be present on the list frame. Frame errors can severely bias sampling and estimation. It is nearly impossible to correct errors in estimates that are based on a sample drawn from a frame with moderate error.[1] In addition to (1) increasing coverage and (2) reducing the number of duplicate records on a list, computerized record linkage models can (3) reduce the number of clerical hours required for review and cleanup and (4) reduce the overall cost of a survey or census.

For its 1987 Census of Agriculture, the Bureau of the Census implemented *ad hoc* algorithms for parsing names and addresses. For pairs of records agreeing on U.S. Postal Zip Code, the software used a combination of (1) surname information, (2) the first character of the given name, and (3) numeric address information to identify 'duplicates' and 'possible duplicates'. Of these pairs of records, (1) 6.6% (396,000) were identified as 'duplicates' and (2) an additional 28.9% (1,734,000) were designated as 'possible duplicates'. The 'possible duplicates' were then reviewed manually. This clerical effort, encompassing about 14,000 h of clerical time over a 3-month period of time, identified an additional 450,000 duplicate records. The Bureau of the Census estimated that about 10% of the records on the final list frame were duplicates. Because there were so many duplicates, some of the estimates calculated from this survey may be substantially in error.

For its 1992 Census of Agriculture, the Bureau of the Census implemented a computerized record linkage model based on the Fellegi–Sunter model and augmented by effective algorithms (see Section *An Empirical Example*) for dealing with typographical errors. The resulting software identified 12.8% of

This report is released to inform interested parties of research and to encourage discussion. The views expressed are those of the authors and not necessarily those of either the U.S. Department of Housing and Urban Development or the U.S. Census Bureau.

*Correspondence to: William.e.winkler@census.gov

[1]National Association of Insurance Commissioners, FHA/HUD, Washington, DC 20410, USA

[2]National Opinion Research Center at the University of Chicago, Chicago, IL, USA

[3]U.S. Census Bureau, Washington DC, USA

DOI: 10.1002/wics.108

the 6-million record file (about 768,000 records) as duplicates and an additional 19.7% as requiring clerical review. This time the number of clerical staff hours was reduced by about half, to 6500 over 22 days, and an additional 486,000 duplicates were identified. Moreover, this time the Bureau of the Census estimated that only about 2% of the records on the final list frame were duplicates.

Another use of record linkage models is to estimate the extent of undercoverage/overcoverage in the U.S. Decennial Census. For both the 1980 and 1990 U.S. Censuses,[2] a large number of census blocks (contiguous regions of approximately 70 households) were re-enumerated. The computerized record linkage model used for the 1990 Census

- reduced the amount of required clerical review and cleanup from an estimated 3000 individuals for 6 months on the 1980 Census to 300 individuals for 6 weeks on the 1990 Census,
- reduced the false match rates from 5.0% on the 1980 Census to approximately 0.2% on the 1990 Census, and
- increased the proportion of matches automatically identified by the computer from 0% on the 1980 Census to more than 85% on the 1990 Census. (Moreover, for the 1990 Census, the remainder of the matches were easily located among potentially matching individuals in the same household. The potentially matching individuals were often missing both first name and age.)

SCOPE OF WORK

In this article, we focus on the record linkage model of Fellegi and Sunter[3] and several of the enhanced practical tools that are needed to handle (often exceptionally) messy data.[a] Although the essence of the approach is statistical, most development has been done by computer scientists using machine learning or database methods.[4] Computer scientists refer to record linkage as *entity resolution, object identification*, or a number of other terms.

Our work proceeds as follows. In Section *The Fellegi–Sunter Model of Record Linkage*, we describe the record linkage model of Fellegi and Sunter.[3] In Sections *Learning Parameters via the Methods of Fellegi and Sunter* and *Learning Parameters via the EM Algorithm*, we present two schemes for estimating the parameters of the Fellegi–Sunter model. The scheme described in Section *Learning Parameters via the Methods of Fellegi and Sunter* requires training data; the more general estimation scheme of Section *Learning Parameters via the EM Algorithm* uses the EM algorithm and does not require the training data. In Section *String Comparators*, we describe string comparator metrics. These are tools that enhance the use of the Fellegi–Sunter model when names of individuals and/or addresses are subject to typographical error. In Section *An Empirical Example*, we present an empirical example. Finally, Section *Summary and Concluding Remarks* consists of concluding remarks.

THE FELLEGI–SUNTER MODEL OF RECORD LINKAGE

Fellegi and Sunter[3] provided a formal mathematical model for ideas that had been introduced by Newcombe et al.[5] and Newcombe and Kennedy.[6] They introduced many ways of estimating key parameters without training data. The methods have been rediscovered in the computer science literature,[7] but without proofs of optimality. To begin, notation is needed. Two files A and B are matched. The idea is to classify pairs in a product space $\mathbf{A} \times \mathbf{B}$ from two files A and B into M, the set of true matches, and U, the set of true nonmatches. Fellegi and Sunter, building rigorous concepts introduced by Newcombe et al.,[5] considered ratios of probabilities of the form:

$$R = P(\gamma \in \Gamma|M)/P(\gamma \in \Gamma|U), \qquad (1)$$

where γ is an arbitrary agreement pattern in a comparison space Γ. For instance, Γ might consist of eight patterns representing simple agreement on the largest name component, street name, and street number. Alternatively, each $\gamma \in \Gamma$ might additionally account for the relative frequency with which specific values of name components such as ‘Smith’, ‘Zabrinsky’, ‘AAA’, and ‘Capitol’ occur. The ratio R or any monotonically increasing function of it, such as the natural log, is referred to as a *matching weight* (or score).

The decision rule is given by:

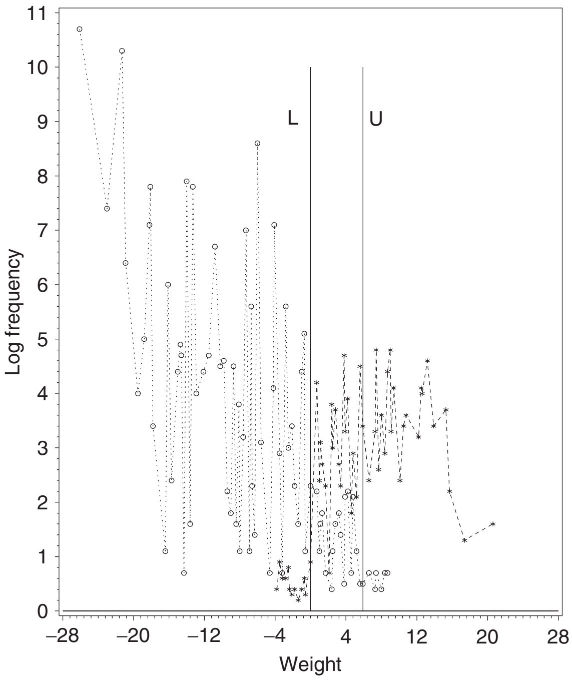
$$\begin{aligned}&\text{If } R > T_\mu, \text{ then designate pair as a match.}\\ &\text{If } T_\lambda \le R \le T_\mu, \text{ then designate pair as a possible}\\ &\quad\text{match and hold for clerical review.}\\ &\text{If } R < T_\lambda, \text{ then designate pair as a nonmatch.} \qquad (2)\end{aligned}$$

The cutoff thresholds T_μ and T_λ are determined by *a priori* error bounds on false matches and false nonmatches. Rule 2 agrees with the intuition. If $\gamma \in \Gamma$ consists primarily of agreements, then it is intuitive that $\gamma \in \Gamma$ would be more likely to occur among matches than nonmatches and ratio 1 would be large.

On the other hand, if $\gamma \in \Gamma$ consists primarily of disagreements, then ratio 1 would be small. Rule 2 partitions the set $\gamma \in \Gamma$ into three disjoint subregions. The region $T_\lambda \leq R \leq T_\mu$ is referred to as the *no-decision region* or *clerical review* region. In some situations, resources are available to review pairs clerically.

Fellegi and Sunter[3] (Theorem 1) proved the optimality of the classification rule given by 2. Their proof is very general in the sense that it holds for any representations $\gamma \in \Gamma$ over the set of pairs in the product space $\mathbf{A} \times \mathbf{B}$ from two files. As they observed, the quality of the results from classification rule 2 was dependent on the accuracy of the estimates of $P(\gamma \in \Gamma|M)$ and $P(\gamma \in \Gamma|U)$.

Figure 1 provides an illustration of the curves of log frequency versus log weight for matches and nonmatches, respectively. The two vertical lines represent the lower and upper cutoffs thresholds T_λ and T_μ, respectively. The *x-axis* is the log of the likelihood ratio R given by 1. The *y-axis* is the log of the frequency counts of the pairs associated with the given likelihood ratio. The plot uses pairs of records from a contiguous geographic region that was matched in the 1990 Decennial Census. The clerical review region between the two cutoffs primarily consists of pairs within the same household, which are missing both first name and age.



o=nonmatch, *=match
cutoff "L" = 0 and cutoff "U" = 6

FIGURE 1 | Log frequency versus weight matches and nonmatches combined.

LEARNING PARAMETERS VIA THE METHODS OF FELLEGI AND SUNTER

Fellegi and Sunter[3] were the first to give very general methods for computing the probabilities in ratio 1. As the methods are useful, we describe what they introduced and then show how the ideas led to more general methods that can be used for *unsupervised learning* (i.e., without training data) in a large number of situations.

Fellegi and Sunter observed several things. First,

$$P(A) = P(A|M)P(M) + P(A|U)P(U), \qquad (3)$$

for any set S of pairs in $\mathbf{A} \times \mathbf{B}$. The probability on the left can be computed directly from the set of pairs. If sets A^x represent simple agreement/disagreement, under the conditional independence assumption (CI), we obtain

$$P(A_1^x \cap A_2^x \cap A_3^x|D) = P(A_1^x|D)P(A_2^x|D)P(A_3^x|D), \quad (4)$$

and then (3) and (4) provide seven equations and seven unknowns (as x represents agree or disagree) that yield quadratic equations they solved. Here D is either M or U. Equation (or set of equations) 4 can be expanded to K fields. Although there are eight patterns associated with the equations of the form 4, we eliminate one because the probabilities must add to one. In general, with more fields but still simple agreement/disagreement between fields, the equations can be solved via the EM algorithm in the next section. Probabilities of the form $P(A_i \mid D)$ are referred to as *m-probabilities* if $D = M$ and *u-probabilities* if $D = U$.

LEARNING PARAMETERS VIA THE EM ALGORITHM

In this section, we do not go into much detail about the basic EM algorithm[8] because the algorithm is well understood. We provide a moderate amount of detail for the record linkage application so that we can describe a number of the limitations of the EM and some of the extensions.

For each $\gamma \in \Gamma$, we consider

$$P(\gamma) = P(\gamma|C_1)P(C_1) + P(\gamma|C_2)P(C_2), \qquad (5a)$$

$$P(\gamma) = P(\gamma|C_1)P(C_1) + P(\gamma|C_2)P(C_2) + P(\gamma|C_3)P(C_3), \qquad (5b)$$

and note that the proportion of pairs having representation $\gamma \in \Gamma$ (i.e., left-hand side of Eq. 5a) can be computed directly from available data. In each of the variants, C_1 and C_2, or C_1, C_2, and C_3 partition $\mathbf{A} \times \mathbf{B}$.

If the number of fields associated with γ is $K > 3$, then we can solve the combination of equations given by 5a and 3 using the EM algorithm. Although there are alternate methods of solving the equation, such as methods of moments and least squares, the EM is greatly preferred because of its numeric stability. Under conditional independence, programming is simplified and computation is greatly reduced (from 2^k to $2k$).

Caution must be observed when applying the EM algorithm to real data. The EM algorithm that has been applied to record linkage is a *latent class algorithm* that is intended to divide $\mathbf{A} \times \mathbf{B}$ into the desired sets of pairs M and U. The probability of a class indicator that determines whether a pair in M or U is the missing data must be estimated along with the m- and u-probabilities. It may be necessary to apply the EM algorithm to a particular subset S of pairs in $\mathbf{A} \times \mathbf{B}$ in which most of the matches M are concentrated, for which the fields used for matching can clearly separate M from U, and for which suitable initial probabilities can be chosen. Because the EM is a local maximization algorithm, the starting probabilities may need to be chosen with care based on experience with similar types of files. Because the EM latent class algorithm is a general clustering algorithm, there is no assurance that the algorithm will divide $\mathbf{A} \times \mathbf{B}$ into two classes C_1 and C_2 that almost precisely correspond to M and U.

The following example characterizes some of the cautions that must be observed when applying the EM. As we will observe, the EM, when properly applied, can supply final limiting parameters that are quite effective. Based on extensive Decennial Census work, the final limiting parameters often reduced the size of the clerical review region by two-thirds from the region that might have been obtained by the initial parameters obtained from knowledgeable guesses. In the following, we use 1988 Dress Rehearsal Census data from one of the 457 regions of the United States that we used for the 1990 Decennial Census. The matching fields consist of last name, first name, house number, street name, phone, age, and sex. In actuality, we also used middle initial, unit (apartment identifier), and marital status. The first file A is a sample of blocks from the region and the second file is an independent enumeration of the same sample of blocks. The first file size is 15,048 and the second file size is 12,072. We only consider 116,305 pairs that agree on Census block ID and first character of surname. A census block consists of approximately 70 households, whereas a ZIP + 4 area represents approximately 50 households. We observe that there can be at most 12,072 matches if the smaller file is an exact subset of the larger file. As is typical in population censuses, the work begins with address lists of households in which the data from the survey forms are used to fill in information associated with individuals. In many situations (such as with families), there will be more than one individual associated with each address (housing unit).

We begin by applying the (2-class) EM to the set of 116,305 pairs. We use knowledgeable initial probabilities that we believe correspond to the probabilities we need for matching individuals. We also use a precursor program to get the counts (or probabilities) of the form $P(\gamma)$ that we use in the EM algorithm. In the limit, we get the final probabilities given in Table 1. The final proportion of matches in the first class $P(M) = 0.2731$ is much too large. The m-probability $P(\text{agree first} \mid M) = 0.31$ is much too small. What has gone wrong? We observe that addresses are of high quality. Because we are in very small contiguous regions (blocks), last name, house number, street name, and phone are likely to be the same in most housing units associated with families. The higher quality household information outweighs the person fields of first name, age, and sex that might be used to separate individuals within household.

We overcome the situation by creating a 3-class EM that we hope divides records agreeing on household variables into two classes and leaves a third class that would be nonmatches outside the households. The initial ideas were due to Smith and Newcombe[9] who provided separate *ad hoc* weighting (likelihood) adjustments for the set of person fields and the set of household fields. As the EM algorithm is quite straightforward to convert to three classes, we make the appropriate algorithmic adjustments and choose appropriate starting probabilities. Winkler[10] provides details. Table 2 gives initial probabilities for a first class that we hope corresponds to person matches M within a household, an in-between class I_h that we hope corresponds to nonmatches within the same household, and a class O_h that are pairs

TABLE 1 | Initial and Final Probabilities from 2-Class EM Fitting

	Initial		Final	
	m	*u*	*m*	*u*
Last	0.98	0.24	0.95	0.07
First	0.98	0.04	0.31	0.01
Hsnm	0.94	0.24	0.98	0.03
Stnm	0.66	0.33	0.99	0.47
Phone	0.70	0.14	0.68	0.01
Age	0.88	0.11	0.38	0.07
Sex	0.98	0.47	0.61	0.49

TABLE 2 | Initial and Final Probabilities from 3-Class EM Fitting

	Initial			Final			
	m	*i*	*oh*	*m*	*i*	*oh*	*u*
Last	0.98	0.90	0.24	0.96	0.92	0.07	0.25
First	0.98	0.24	0.04	0.96	0.02	0.01	0.01
Hsnm	0.94	0.90	0.24	0.97	0.97	0.04	0.23
Stnm	0.66	0.90	0.33	0.98	0.99	0.47	0.58
Phone	0.70	0.60	0.14	0.72	0.64	0.01	0.14
Age	0.88	0.20	0.11	0.88	0.14	0.07	0.08
Sex	0.98	0.70	0.47	0.98	0.45	0.49	0.49

not agreeing on household fields. To get the final u-probabilities, we combine the i-probabilities and the oh-probabilities according to the proportions in classes 2 and 3. When we run the EM program, we get probabilities of being in the three classes of 0.0846, 0.1958, and 0.7196, respectively. The probability 0.0846 associated with the first class accurately corresponds to the known number of true matches (obtained via two levels of review and one level of adjudication). The starting i-probabilities are reasonable guesses for the probabilities of persons within the same household who are not matches.

If the EM algorithm is applied with care, then it will generally yield good parameter estimates with lists of individuals. It will not always yield reasonable parameters with agriculture or business lists because of the (moderately) high proportion of truly matching pairs that disagree on names or addresses. The EM algorithm was used for production matching in the 1990 Decennial Census[2,4] because Winkler had been able to demonstrate that matching probabilities (particularly m-probabilities) varied significantly (say between a suburban area and an adjacent urban area). If we think of $1 - P(A_i \mid M)$ as crudely representing the average typographical error in the ith field, then the variation of parameters is understandable because lists associated with urban areas often contain more typographical error.

Winkler[10,11] showed that the EM algorithm yielded 'optimal parameters' in the sense of effective local maxima of the likelihood. The 2-class and 3-class EM algorithms under condition (CI) are quite robust. If starting points are varied substantially, the EM converges to the same limiting values where the limiting values are determined by characteristics of the files A and B. The 2-class algorithm will outperform the 3-class algorithm in situations where there is typically only one entity at an address (or telephone number). In those situations, the address can be considered as an identifier of the individual entity.

During 1990 production matching, the EM algorithm showed its flexibility. In three regions among a number of regions processed in 1 week, clerical review became much larger with the EM parameters than was expected. Upon quick review, supervisors determined that two keypunchers had managed to bypass edits on the year of birth. All records from these keypunchers disagreed on the computed age. The clerical review became much larger because first name and the age were the main fields for separating persons within a household.

Ravikumar and Cohen[12] and Bhattacharya and Getoor[13] provide unsupervised methods of learning parameters that generalize the EM methods of Winkler[11] and are related to the general methods of Winkler.[10]

STRING COMPARATORS

In most matching situations, we will get poor matching performance when we compare two strings exactly (character-by-character) because of typographical error. Dealing with typographical error via approximate string comparison has been a major research project in computer science (see, e.g., Refs 14 and 15). In record linkage, we need to have a function that represents approximate agreement, with agreement being represented by 1 and degrees of partial agreement being represented by numbers between 0 and 1. We also need to adjust the likelihood ratios 1 according to the partial agreement values. Having such methods is crucial to matching. For instance, in a major census application for measuring undercount, more than 25% of matches would not have been found via exact character-by-character matching. Three geographic regions (St Louis—urban, Columbia, MO—suburban, and Washington—suburban/rural) are considered in Table 3. The function Φ represents exact agreement when it takes value 1 and represents partial agreement when it takes values <1. In the St Louis region, for instance, 25%

TABLE 3 | Proportional Agreement by String Comparator Values Among Matches

	St Louis	Columbia	Washington
First			
$\Phi = 1.0$	0.75	0.82	0.75
$\Phi \geq 0.6$	0.93	0.94	0.93
Last			
$\Phi = 1.0$	0.85	0.88	0.86
$\Phi \geq 0.6$	0.95	0.96	0.96

Key fields by geography.

of first names and 15% of last names did not agree character-by-character among pairs that are matches.

Jaro[16] introduced a string comparator that accounts for insertions, deletions, and transpositions. The basic Jaro algorithm has three components: (1) compute the string lengths, (2) find the number of common characters in the two strings, and (3) find the number of transpositions. The definition of common is that the agreeing character must be within half the length of the shorter string. The definition of transposition is that the character from one string is out of order with the corresponding common character from the other string. The string comparator value (rescaled for consistency with the practice in computer science) is:

$$\Phi_j(s_1, s_2) = \frac{1}{3}\left(\frac{N_C}{\text{len}_{s_1}} + \frac{N_C}{\text{len}_{s_2}} + \frac{0.5N_t}{N_C}\right), \qquad (6)$$

where s_1 and s_2 are the strings with lengths len_{s1} and len_{s2}, respectively, N_C is the number of common characters between strings s_1 and s_2 where the distance for common is half of the minimum length of s_1 and s_2, and N_t is the number of transpositions. The number of transpositions N_t is computed somewhat differently from the obvious manner.

Using truth data sets, Winkler[17] introduced methods for modeling how the different values of the string comparator affect the likelihood 1 in the Fellegi–Sunter decision rule. Winkler[17] also showed how a variant of the Jaro string comparator Φ dramatically improves matching efficacy in comparison with situations when string comparators are not used. The Winkler variant uses some ideas of Pollock and Zamora[18] in a large study for the Chemical Abstracts Service. They provided empirical evidence that quantified how the probability of keypunch errors increased as the character position in a string moved from the left to the right. The Winkler variant, referred to as the *Jaro–Winkler string comparator*, is widely used in computer science.

Work by Cohen et al.[19,20] provides empirical evidence that the new string comparators can perform favorably in comparison with bigrams and edit distance. Edit distance uses dynamic programming to determine the minimum number of insertions, deletions, and substitutions to get from one string to another. The bigram metric counts the number of consecutive pairs of characters that agree between two strings. A generalization of bigrams is q-grams, where q can be greater than 2. Cohen et al.[19,20] provided additional string comparators that they demonstrated slightly outperformed the Jaro–Winkler string comparator with several small test decks, but not with a test deck similar to Census data. Yancey,[21] in a rather exhaustive study, also demonstrated that Jaro–Winkler string comparator outperformed new string comparators of Cohen et al.[19,20] with large census test decks. Yancey introduced several hybrid string comparators that used both the Jaro–Winkler string comparator and variants of edit distance.

Table 4 compares the Jaro, Winkler, bigram, and edit distance values for selected first names and last names. Bigram and edit distance are normalized to be between 0 and 1. All string comparators take value 1 when the strings agree character-by-character.

AN EMPIRICAL EXAMPLE

In the following, we compare different matching procedures on the data that were used for the initial EM analyses (Tables 1 and 2). Although we also demonstrated very similar results with several alternative pairs of files, we do not present the additional results here.[17] The results are based only on pairs that agree on block identification code and first character of the last name.

The procedures that we use are as follows. The simplest procedure, *crude*, merely uses an *ad hoc* (but knowledgeable) guess for matching parameters and does not use string comparators. The next, *param*, does not use string comparators but does estimate the m- and u-probabilities. Such probabilities are estimated through an iterative procedure that involves manual review of matching results and successive reuse of re-estimated parameters. Such iterative refinement procedures are a feature of Statistics Canada's CANLINK system.

The third type, *param2*, uses the same probabilities as *param* and the basic Jaro string comparator. The fourth type, *em*, uses the EM algorithm for estimating parameters and the Jaro string comparator. The fifth type, *em2*, uses the EM algorithm for estimating parameters and the Winkler variant of the string comparator that performs an upward adjustment based on the amount of agreement in the first four characters in the string.

In Table 5, the cutoff between designated matches is determined by a 0.002 false match rate. The *crude* and *param* types are allowed to rise slightly above the 0.002 level because they generally have higher error levels. In each pair of columns (designated matches and designated clerical pairs), we break out the counts into true matches and true nonmatches. In the designated matches, true nonmatches are false matches.

By examining the table, we observe that a dramatic improvement in matches can occur when

TABLE 4 | Comparison of String Comparators Using Last Names and First Names

		String Comparator Values			
Two Strings		Jaro	Winkler	Bigram	Edit
SHACKLEFORD	SHACKELFORD	0.970	0.982	0.800	0.818
DUNNINGHAM	CUNNIGHAM	0.867	0.867	0.917	0.889
NICHLESON	NICHULSON	0.926	0.956	0.667	0.889
JONES	JOHNSON	0.867	0.893	0.167	0.667
MASSEY	MASSIE	0.889	0.933	0.600	0.667
ABROMS	ABRAMS	0.889	0.922	0.600	0.833
HARDIN	MARTINEZ	0.778	0.778	0.286	0.143
ITMAN	SMITH	0.467	0.467	0.200	0.000
JERALDINE	GERALDINE	0.926	0.926	0.875	0.889
MARHTA	MARTHA	0.944	0.961	0.400	0.667
MICHELLE	MICHAEL	0.833	0.900	0.500	0.625
JULIES	JULIUS	0.889	0.933	0.800	0.833
TANYA	TONYA	0.867	0.880	0.500	0.800
DWAYNE	DUANE	0.778	0.800	0.200	0.500
SEAN	SUSAN	0.667	0.667	0.200	0.400
JON	JOHN	0.778	0.822	0.333	0.750
JON	JAN	0.778	0.800	0.000	0.667

TABLE 5 | Matching Results via Matching Strategies

	Designated Computer Match	Designated Clerical Pair
Truth	Match/Nonmatch	Match/Nonmatch
crude	310/1	9344/794
param	7899/16	1863/198
param2	9276/23	545/191
em	9587/23	271/192
em2	9639/24	215/189

0.2% False matches among designated matches.

string comparators are first used (from *param* to *param2*). The reason is that disagreements (on a character-by-character basis) are replaced by partial agreements and adjustment of the likelihood ratios.[17] The improvement due to the Winkler variant of the string comparator (from *em* to *em2*) is quite minor. The *param* method is essentially the same as a traditional method used by Statistics Canada. After a review of nine string comparator methods,[22] Statistics Canada provided options for three string comparators in CANLINK software with the Jaro–Winkler comparator being the default.

The improvement between *param2* and *em2* is not quite as dramatic because it is much more difficult to show improvements among 'hard-to-match' pairs and because of the differences in the parameter estimation methods. Iterative refinement is used for *param* and *param2* (a standard method in CANLINK software) in which an appropriate subset of pairs is reviewed, reclassified, and parameters re-estimated. This method is a type of (partially) supervised learning and is both labor-intensive and time-consuming. The parameter estimation variants of Table 5 have consistently shown greater improvement with other pairs of files.

The improvement due to the parameters from *em* and *em2* can be explained because the parameters are slightly more general than those obtained under conditional independence (*param2*). If A_i^x represents agreement or disagreement on the *i*th field, then the conditional independence assumption yields

$$P(A_1^x \cap A_2^x \cdots \cap A_k^x | D) = \prod_{i=1}^{k} P(A_i^x | D), \qquad (7)$$

where D is either M or U. Superficially, the EM considers different orderings of the form

$$P(A_{\rho,1}^x \cap \cdots \cap A_{\rho,k}^x | D) = \prod_{i=1}^{k} P(A_{\rho,i}^x | A_{\rho,i-1}^x, \ldots, A_{\rho,1}^x, D), \qquad (8)$$

where ρ,i represents the ith entry in a permutation ρ of the integers 1 to k. The greater generality of 8 in comparison with 7 can yield better fits to the data. We can reasonably assume that the EM algorithm under the conditional independence assumption (as the actual computational methods work) simultaneously chooses the best permutation ρ and the best parameters.

In this section, we have demonstrated the very dramatic improvement in record linkage efficacy through advancing from seemingly reasonable *ad hoc* procedures to procedures that use modern computerized record linkage. The issue that affects statistical agencies is whether their survey frames are well maintained using effective procedures. Upgrading matching procedures is often as straightforward as replacing a subroutine that uses *ad hoc* methods with another subroutine.

SUMMARY AND CONCLUDING REMARKS

In this article, we have discussed modern computerized record linkage procedures that are used for (1) removing duplicate entries from sampling frames or business lists, (2) improving the coverage (or completeness) of such frames or lists, and (3) estimating the extent of undercoverage/overcoverage in a population census. We began by giving a brief definition of record linkage and describing a few of its applications. We then presented the general form of the Fellegi–-Sunter record linkage model as well as two schemes for estimating the parameters of the model. We went on to discuss string comparators metrics. These are practical enhancements to the Fellegi–Sunter model that treat typographical errors. We concluded the entry with an empirical example.

We find it remarkable that modern computerized record linkage methods can lead to substantial improvements in the quality of lists at substantially reduced cost and in a more timely fashion.

NOTES

[a] This article is necessarily a brief summary of the topic of record linkage. For a more complete discussion, including many real-world examples, the interested reader should see Ref 23.

REFERENCES

1. Deming WE, Gleser GJ. On the problem of matching lists by samples. *J Am Stat Assoc* 1959, 54:403–415.
2. Winkler WE. Matching and record linkage. In: Cox BG, Binder DA, Chinnappa BN, Christianson A, Colledge MJ, Kott PS, eds. *Business Survey Methods*. New York: John Wiley & Sons; 355–384. Available at: http://www.fcsm.gov/working-papers/wwinkler.pdf. Accessed 1995.
3. Fellegi IP, Sunter AB. A theory for record linkage. *J Am Stat Assoc* 1969, 64:1183–1210.
4. Winkler WE. *Overview of record linkage and current research directions*, 2006., Available at: http://www.census.gov/srd/papers/pdf/rrs2006-02.pdf. Accessed 2006.
5. Newcombe HB, Kennedy JM, Axford SJ, James AP. Automatic linkage of vital records. *Science* 1959, 130:954–959.
6. Newcombe HB, Kennedy JM. Record linkage: Making maximum use of the discriminating power of identifying information. *Commun Assoc Comput Mach* 1962, 5:563–567.
7. Cooper WS, Maron ME. Foundations of probabilistic and utility-theoretic indexing. *J Assoc Comput Mach* 1978, 25:67–80.
8. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc, B* 1977, 39:1–38.
9. Smith ME, Newcombe HB. Methods of computer linkage for hospital admission–separation records into cumulative health histories'. *Methods Inf Med* 1975, 14:18–25.
10. Winkler WE. Improved decision rules in the Fellegi-Sunter model of record linkage. In: *Proceedings of the Section on Survey Research Methods*. American Statistical Association, Alexandria, VA; 1993, 274–279.
11. Winkler WE. Using the EM algorithm for weight computation in the Fellegi-Sunter model of record linkage. In: *Proceedings of the Section on Survey Research Methods*. American Statistical Association, Alexandria, VA; 1988, 667–671.
12. Ravikumar P, Cohen WW. A hierarchical graphical model for record linkage. *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Banff, Calgary, CA, July 2004.
13. Bhattacharya I, Getoor L. *A latent dirichlet allocation model for entity resolution*. SIAM Data Mining 06—best paper, 2006.

14. Hall PAV, Dowling GR. Approximate string comparison. *Assoc Comput Mach, Comput Surv* 1980, 12:381–402.

15. Navarro G. A guided tour of approximate string matching. *Assoc Comput Mach, Comput Surv* 2001, 33:31–88.

16. Jaro MA. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *J Am Stat Assoc* 1989, 89:414–420.

17. Winkler WE. String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage. In: *Proceedings of the Section on Survey Research Methods*. American Statistical Association, Alexandria, VA; 1990, 354–359.

18. Pollock J, Zamora A. Automatic spelling correction in scientific and scholarly text. *Commun ACM* 1984, 27:358–368.

19. Cohen WW, Ravikumar P, Fienberg SE. A comparison of string metrics for matching names and addresses. *International Joint Conference on Artificial Intelligence, Proceedings of the Workshop on Information Integration on the Web*, Acapulco, Mexico, August 2003.

20. Cohen WW, Ravikumar P, Fienberg SE. A comparison of string distance metrics for name-matching tasks. *Proceedings of the ACM Workshop on Data Cleaning, Record Linkage and Object Identification*, Washington DC, August 2003.

21. Yancey WE. *Evaluating string comparator performance for record linkage. Research Report RRS 2005/05*, 2005, Available at: http://www.census.gov/srd/www/byyear.html. Accessed 2005.

22. Budzinsky CD. Automated spelling correction. *Statistics Canada Technical Report*, Statistics Canada, Ottawa, Ontario, Canada; 1991.

23. Herzog TN, Scheuren F, Winkler WE. *Data Quality and Record Linkage*. New York: Springer; 2007.