ıter Record Linkage Model

uency-based estimates

ɔach	Total
Non-match	
2	328
0	157
2,184	2,292
2,186	2,777

e frequency-based matching
ches versus 157 for the basic
had (1) a relatively rare first
if any, agreements on other

schemes for estimating the
odel. We also discussed the
concluded the chapter with a
based record linkage to basic
; a number of techniques that

10
Standardization and Parsing

The purpose of record linkage, as noted in previous chapters, is to find (1) pairs of records across files that correspond to the same entities (e.g., individuals or businesses) or (2) duplicates within a single file. If unique identifying numbers are not available, then the name and/or address are usually used. These names and addresses can be considered to be strings of characters or *character strings*.

The first step in comparing such character strings is to identify commonly used terms such as road, street, mister, or junior. These are then converted, if necessary, to a standard form or abbreviation. We use *standardization* to refer to this conversion of common terms to a standard form or abbreviation. Common terms such as Doctor, Missus, etc. (Dr, Mrs.) or Street, Drive, etc., or Corporation, Company etc. (Corp., Co.) can be relatively easy to find and standardize.

The second step is to partition the character string, into its component parts and to place these parts into identifiable fields. In the terminology of record linkage, we use the term *parse* to indicate this type of partition. When parsing these strings it is often useful to make use of the results of the standardization process. Specifically, the standardized words can serve as keywords into special routines for parsing. For instance, if you find the term "Company" in the string, then you are likely dealing with the name of a business and so want to call on a computer subroutine that parses business addresses. Similarly, Post Office Box is different from Rural Route Box Number; and Rural Route Box Number is different from House Number Street Name.

Standardization and parsing are used to facilitate matching and need to be completed before comparing the records for record linkage purposes.

Standardization is discussed in Section 10.2, parsing in Section 10.3. Unfortunately, it is not always possible to compare/match two strings exactly (character-by-character) because of typographical error and/or variations in spelling. In record linkage, we need a function that represents approximate agreement, with complete agreement being represented by 1 and partial agreement being represented by a number between 0 and 1. We also need to incorporate these partial agreement values into the weighting process – that is, to adjust the weights (likelihood ratios of the Fellegi–Sunter scheme). Such functions are called *string comparator metrics* and are discussed in Chapter 13. Additionally, we sometimes encode street names, surnames, and/or given names to facilitate comparison. Two

TABLE 10.1. Which pairs represent the same person?

Name	Address	Age
John A Smith	16 Main Street	16
J H Smith	16 Main St	17
Javier Martinez	49 E Applecross Road	33
Havier Matreenez	49 Aplecross Raod	36
Bobbie Sabin	645 Reading Aev	22
Roberta Sabin	123 Norcross Blvd	
Gillian Jones	645 Reading Aev	24
Jilliam Brown	123 Norcross Bvd	43

widely used coding schemes, SOUNDEX and NYSIIS, used for such purposes, are described in Chapter 11. In Chapter 12 we describe a scheme known as blocking that reduces the number of record pairs we need to compare.

We begin this chapter by looking at some examples of the problems for which standardization and parsing are likely to be helpful.

Example 10.1: Which pairs are true?

We consider the following pairs of entries taken from a variety of lists (Table 10.1).

The question that arises is, Which of the pairs above represent the same person? The answer is that we need to see each of the lists in its entirety. We need to know the context in which the entries appear. For instance, with the Sabin example, the first entry might be from a list of individuals in a college glee club and the second might be from a list of known graduates of the college. Concerning the Jones/Brown example, the first entry might be from a list of medical students at a particular university 20 years ago and the second list might be a current list of practicing physicians who had graduated from the university's medical school. Relatively inexperienced clerical staff should be able to figure out the first two examples easily. We would like to have computer software that does some of the things a person can do and also does the things that a computer does well.

Example 10.2: Which pairs are real, again?

We consider the names on two records as shown in Table 10.2

TABLE 10.2. Names as originally entered on data record

Record	Name
I	Smith, Mr. Bob, Jr.
II	Robert Smith

TABLE 10.3. Standardized names

Record	Standardized name
I	Smith, Mr. Robert, Jr.
II	Robert Smith

Each of the names can be considered to be a single string. The standardized versions of these names are shown in Table 10.3.

Again, each of the standardized names is simply a string. If we then parse the standardized names, we might end up with the following:

TABLE 10.4. Parsed and standardized names

Record	Prefix	First name (standardized)	Surname	Suffix
I	Mr.	Robert	Smith	Jr.
II		Robert	Smith	

We analyze the situation as follows. We began with the names "Smith, Mr. Bob Jr." and "Robert Smith" appearing on two records (call them I and II, respectively). "Smith, Mr. Bob Jr." is composed of a last name "Smith," a title "Mr.", an abbreviation of a first name "Bob," and a suffix, "Jr.". "Robert Smith" is composed of a (unabbreviated) first name, "Robert", and a last name "Smith." Without the ability to identify the different parts of a name and to place standard spellings or abbreviations for these parts into fixed fields, one could only compare the two character strings on a letter-by-letter basis to see if they were identical. Comparing "S" to "R," "m" to "o," "j" to "b," etc., would cause the computer to believe that the two name fields were different. With the ability to parse and standardize the names, the two name fields would appear as shown in Table 10.4. The computer could then compare each corresponding part. It would discover that it had two missing values and two perfect agreements, rather than a disagreement on a single long string.

10.1. Obtaining and Understanding Computer Files

What files can we use for research studies? What work needs to be done so that the information on a file can be used for matching? What work needs to be done so that we can match records across files? We need an annotated layout. We need to know which records are "in-scope," as some files contain copies of records with a previous address and a status code identifying such records as duplicates. We need to determine the proportion of records that is blank.

Observation: Prior to matching, files must be put in common forms (made homogenous) so that corresponding fields can be compared. It can take more

time (moderately skilled human intervention) to pre-process a small local list than it does to pre-process a very large, well-documented, well-maintained national list.

Rule of Thumb: Get the fewest lists possible to use in updating or creating a merged file. Record linkage errors are often cumulative. If a record is erroneously contained in a file (because it is out of scope or a duplicate), then it may be added to another file (during updating) and increase error (duplication) in the updated file.

10.2. Standardization of Terms

Before a character string (such as a name or an address) is parsed into its components parts and these parts are placed into identifiable fields, each of these fields may need to be *standardized* or converted into a standard form or abbreviation. This *standardization* process (also known as data cleansing or attribute-level reconciliation) is used before performing record linkage in order to increase the probability of finding matches. Without standardization, many true matches would be erroneously designated as non-matches because the common identifying attributes would not have sufficient similarity.

The basic ideas of standardization are concerned with standardization of spelling, consistency of coding, and elimination of entries that are outside of the scope of the area of interest.

10.2.1. Standardization of Spelling

Replace spelling variations of commonly occurring words with a common consistent spelling. For example, replace "Doctor" or "Dr" by "Dr"; replace nicknames such as "Bob" and "Bill" by "Robert" and "William," respectively; replace "rd" or "road" by "rd"; and replace "company," "cmpny," or "co" by "co." We note that the last example is dependent on the context of the term as "co" might refer to county or even Colorado.

10.2.2. Consistency of Coding

Standardize the representation of various attributes to the same system of units or to the same coding scheme. For example, use 0/1 instead of M/F for a "gender" attribute; replace "January 11, 1999" or "11 January 1999" by "01111999" (i.e., MMDDYYYY) or "19990111" (i.e., YYYYMMDD).

10.2.3. Elimination of Entries That Are Out of Scope

If a list of registered voters contains entries for individuals that are deceased or relocated, these need to be edited in an appropriate fashion – that is, by adding status flags or codes, as required.

An electric utility company list of customers is a good source of residential addresses. However, such a list may contain some commercial addresses as well. So, if the goal is to compile a list of residential addresses, it is necessary to eliminate the out-of-scope commercial addresses from the source list.

Next, we consider a case involving two lists. The first is a list of the residents of a small, contiguous portion of a large urban area, while the second list is that of the entire urban area. We wish to extract telephone numbers from the larger list to augment the information on the smaller list. Before linking records, we want to eliminate the records on the larger list that are not in the geographic area represented by the smaller list. This might be accomplished by using the Zip Codes on the record entries.

10.2.4. Perform integrity checks on attribute values or combinations of attribute values

In this regard, we can use the types of schemes described earlier in Chapter 5.

10.2.5. Final Thoughts on Standardization

Standardization methods need to be specific to the population under study and the data collection processes. For example, as noted earlier, the most common name misspellings differ based upon the origin of the name. Therefore, standardization for Italian names optimized to handle Italian names and Latin origin will perform better than generic standardization.

10.3. Parsing of Fields

It is not easy to compare free-form names and addresses except possibly manually. Parsing partitions a free-form string (usually a name or an address) into a common set of components that can be more easily compared by a computer. Appropriate parsing of name and address components is the most critical part of computerized record linkage. Parsing requires the identification of the starting and ending positions of the individual components of the string. For names, we usually need to identify the locations of the first name, middle initial, and last name. For addresses, we frequently need to identify the locations of the house number and the street name. Without effectively parsing such strings, we would erroneously designate many true matches as non-matches because we could not compare common identifying information. For specific types of establishment lists, the drastic effect of parsing failure has been quantified (see Winkler [1985b and 1986]). DeGuire [1988] presents an overview of ideas needed for parsing (as well as standardizing) addresses. Parsing of names requires similar procedures.

TABLE 10.5. Examples of name parsing

Standardized name	Parsed							
	Pre	First	Middle	Last	Post1	Post2	Bus1	Bus2
Dr. John J Smith MD	DR	John	J	Smith	MD			
Smith DRY FRM				Smith			DRY	FRM
Smith & Son ENTP				Smith		Son	ENTP	

10.3.1. *Parsing Names of Individuals*

In the examples of Table 10.5, the word "Smith" is the name component with the most identifying information. "PRE" refers to a prefix, "POST1" and "POST2" refer to postfixes, while "BUS1" and "BUS2" refer to commonly occurring words associated with businesses. While exact, character-by-character comparison of the standardized but unparsed names would yield no matches, use of the subcomponent last name "Smith" might help to designate some pairs as matches. Parsing algorithms are available that can deal with either last-name-first types of names such as "Smith, John" or last-name-last types such as "John Smith." None are available that can accurately parse both types of names within a single file.

More generally, in order to make the matching of records on individuals efficient, we need high-quality, time-independent identifiers on these individuals. These identifiers include given name, middle initial, last name, maiden name (if appropriate), Social Security number, date of birth (preferably in the format of MMDDYYYY), and city of birth.

10.3.2. *Parsing of Addresses*

Humans can easily compare many types of addresses because they can associate corresponding components in free-form addresses. To be most effective, matching software requires corresponding address subcomponents in specified locations. As the examples in Table 10.6 show, parsing software partitions a free-form address into a set of components each of which is in a specified location.

TABLE 10.6. Examples of address parsing

Standardized address	Parsed									
	Pre2	HSNM	STNM	RR	Box	Post1	Post2	Unit1	Unit2	Bldg
16 W Main ST APT 16	W	16	Main			ST		16		
RR 2 BX 215				2	215					
Fuller BLDG SUITE 405									405	Fuller
14588 HWY 16 W		14588	Hwy 16				W			

Parsed				
ast	Post1	Post2	Bus1	Bus2
mith	MD			
mith			DRY	FRM
mith		Son	ENTP	

luals

th" is the name component with the
o a prefix, "POST1" and "POST2"
refer to commonly occurring words
aracter-by-character comparison of
yield no matches, use of the sub-
) designate some pairs as matches.
with either last-name-first types of
t types such as "John Smith." None
ypes of names within a single file.
atching of records on individuals
lent identifiers on these individuals.
initial, last name, maiden name (if
f birth (preferably in the format of

ddresses because they can associate
ddresses. To be most effective,
ddress subcomponents in specified
how, parsing software partitions a
s each of which is in a specified

Parsed					
Box	Post1	Post2	Unit1	Unit2	Bldg
	ST		16		
215					
				405	Fuller
		W			

TABLE 10.7. Pairs of names referring to the same business entity

Name	Explanation
John J Smith ABC Fuel Oil	One list has the name of the owner while the other list has the name of the business.
John J Smith, Inc J J Smith Enterprises	These are alternative names of the same business.
Four Star Fuel, Exxon Distributor Four Star Fuel	One list has both the name of the independent fuel oil dealer and the associated major oil company.
Peter Knox Dairy Farm Peter J Knox	One list has the name of business while the other has the name of the owner.

TABLE 10.8. Names referring to different businesses

Name	Explanation
John J Smith Smith Fuel	Similar names but different companies
ABC Fuel ABC Plumbing	Identical initials but different companies
North Star Fuel, Exxon Distributor Exxon	Independent affiliate and company with which affiliated

10.3.3. Parsing Business Names

The main difficulty with business names is that even when they are parsed correctly, the identifying information may be indeterminate. In each example of Table 10.7, the pairs refer to the same business entities that might be in a list frame constructed for a sample survey of businesses.

In Table 10.8 each pair refers to different business entities that have similar components.

Because the name information in Tables 10.7 and 10.8 may not be sufficiently accurate to determine match status, address information or other identifying characteristics may have to be obtained via clerical review. If the additional address information is indeterminate, then at least one of the establishments in each pair may have to be contacted.

What information do we need to match individual business enterprises efficiently? We need information such as the business's name, the Zip Code of the business's headquarters, the Standard Industrial Classification (SIC) Code, or

North American Industry Classification System (NAICS) code of the business, as well as additional quantitative information.

10.3.4. Ambiguous Addresses

A postal address should not merely be regarded from a syntactic point of view. Its semantic content (i.e., its meaning) must be examined as well. Sometimes, a recorded address could potentially represent two or more physical locations. In order to resolve this potential ambiguity more knowledge may be required to exclude non-existent locations and to determine the correct location. Unfortunately, this does not always lead to a resolution and so we might still end up with an ambiguity that we cannot resolve. Consider the following address:

61 1 4-th Street, N.W., Washington, D.C.

Does this represent "611 4-th Street, N.W." or "61 14-th Street, N.W."? What about the address

976 Fort St John BC?

Which of the following does the last address represent:

Apt 976, Fort St-John, BC,
Apt 976 Fort, St-John, BC, or
Apt 976 Fort ST, John, BC.?

10.3.5. Concluding Thought on Parsing

Finally, no matter how good the software becomes, the unsolvable and the non-existent addresses will remain a problem and should be followed up manually.

10.4. Where Are We Now?

In this chapter we discussed two techniques – standardization and parsing – that can be used to enhance record linkages. In Chapters 11–13, we discuss other techniques that also enhance record linkages.