FORMING ESTABLISHMENT UNITS IN THE INTERNATIONAL PRICE PROGRAM

James A. Himelein, Jr. and James F. Carpenter, Bureau of Labor Statistics

I. INTRODUCTION

The International Price Program (IPP) produces quarterly price indexes of products exported from the United States. A sample of exporters is drawn from a frame that is constructed from Shippers Export Declarations which, as prescribed by law, are filed with the Census Bureau by exporters or their agents. The IPP collects the product pricing information at the establishments of the exporters.

The primary sampling unit is an establishment or related set of establishments which belong to an exporting firm or individual. The sampling units are formed by matching the exporter name and address information that was recorded on the individual Shippers Export Declarations and then coding each declaration as part of a particular exporter. This process is called EXPORTER CODING.

The current CODING procedure requires considerable time and effort because of the extensive visual inspection of computer listings and online files that is needed to determine the matches.

This paper presents the research which was undertaken to find ways to improve the CODING process. The outcome of the research was the development and testing of a computer program which automates the coding of the declaration records.

II. BASIC CONCEPTS AND DEFINITIONS

Throughout the following discussion, a record corresponds to a unique name and address alias obtained from the set of declarations.

The CODING process is primarily a matching procedure. Two records are defined to match if their names and addresses represent one or more establishments of the same exporter within a local area (possibly a metropolitan area or a smaller local municipality). The ESTABLISHMENT UNITS are defined as the EXPORTER GROUPS which are formed from the sets of matching records.

We defined a distinction between the AUTO CODING and VISUAL CODING PROCESSES. In VISUAL CODING, the matching decisions have been made manually by trained staff who were aided by auxilliary software. The AUTO CODING is a computer program that automatically makes and implements the matching decisions without manual intervention. In the rest of the paper, the existing IPP procedure is defined as the VISUAL CODING process and the new software as the AUTO CODING. The terms VISUAL and AUTO GROUPS identify from which process the EXPORTER GROUPS have been formed.

III. RESULTS

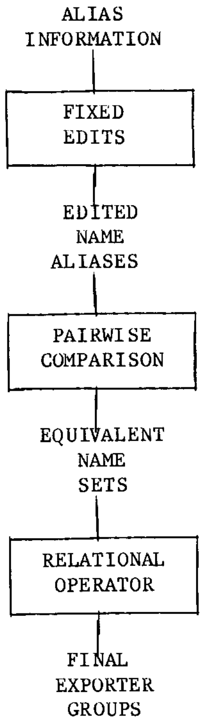
The results were very encouraging. When tested, the AUTO CODING software identified up to 82% of all valid matches with less than a 5% error rate. The basic approach proved to be sound and the experience to be gained from future coding runs will enable us to improve on these results.

IV. METHODS

SUMMARY

The AUTO CODING process has three basic operations. These are:

1. FIXED EDITS
   Each name alias was tested against a fixed set of EDIT rules. The purpose of the EDITS was to standardize common words into shorter abbreviations and remove the minor words which should not have any effect on the matching decisions. The output of this stage was alias records with EDITED names.
2. PAIRWISE COMPARISON
   Pairs of EDITED NAME alias records were compared. A similarity measure for a given pair was calculated. If the measure exceeded the predetermined critical level, then the given name aliases were defined to be equivalent.
3. RELATIONAL OPERATOR
   Within the individual sets of records that had equivalent names, each pair of records was tested for a match. The EXPORTER GROUPS were formed from the partition defined by the equivalence relationship that was extended from the pairwise matches.



DETAILED METHODS DESCRIPTION

FIXED EDITS

The edit rules were applied only to the NAME information as follows:

i. When 'THE' appeared as the first word, it was removed from the EDITED name.

ii. Leading words common to many firms ('AMERICAN', 'GENERAL', 'NATIONAL', 'INTERNATIONAL') were given standard abbreviations ('AMER', 'GEN', 'NATL', 'INTL'). By doing this, we tried to avoid missing equivalences because of the use of abbreviations, as well as, minimizing the risk of making false equivalences due to relatively long common leading words.

iii. Words such as COMPANY, LIMITED, and INCORPORATED which provided no real discriminatory information were deleted from the final EDITED name.

iv. The word 'AND' was given a one character abbreviation, '&'.

v. All punctuation and blanks were deleted. The resulting EDITED name was a continguous character string.

PAIRWISE COMPARISON

The purpose of the PAIRWISE COMPARISON procedure is to determine matches that otherwise would not have been made because of slight variations in the NAME information. Such variations included different word endings, inconsistent abbreviations, and minor transcription errors.

The PAIRWISE COMPARISON was carried out in two steps. First, a similarity measure was calculated for pairs of NAME aliases. The similarity measure quantified the degree of agreement between the pair of aliases. In the second step, the measures were tested against predetermined critical levels. When the measures met or exceeded the critical levels, the NAME aliases were defined to be equivalent.

The similarity measure, denoted as the M VALUE, was calculated as follows.

I. Two edited NAME strings were compared. One of the strings was designated as the SHORTER STRING.

II. For each character of the SHORTER STRING, the LONGER STRING was searched from the point of the last common character (or beginning for the initial letter) through the end character. The common substring found after one cycle of searches defined the KEYSTRING.

III. The length of the KEYSTRING defined the M VALUE.

EXAMPLES

SHORTER STRING	LONGER STRING	KEYSTRING	M VALUE
BEGIN	BENIGN	BEGN	4
INTL	INTERNATIONAL	INTL	4
VARIANCE	VARIATION	VARIAN	6
GOPHER	SOPHOMORE	OPHE	4

Initially, the critical levels were set based upon an analysis of proportions of pairs found to be VISUALLY matched within categories defined by SHORTER STRING length amd M values. When we began testing, we found that very short strings led to increased numbers of erroneous name equivalences and FALSE matches. Thus, the final criteria were graduated as follows:

LENGTH OF SHORTER STRING		CRITICAL
MINIMUM	MAXIMUM	M-VALUE
1	4	3
5	7	4
8	9	5
10	11	6
12	13	7
14	36	8

To facilitate programming we decided that only adjacent pairs of records would be put through the PAIRWISE COMPARISON operation. Thus, the determination of the records to be compared was made by the sort of the record file. In the final test, the record files were sorted alphabetically by EDITED NAME.

We had hoped that blocks of equivalent NAME records would fall together. In initial testing, we found that some care had to be taken when linking the names into equivalent blocks.

Suppose records A, B, and C are compared in that order. A and B are compared and it is found that NAME(A) = NAME(B). Likewise B and C are compared and NAME(B) = NAME(C). It seemed natural to let the comparisons link and define NAME(A) = NAME(B) = NAME(C). However, if NAME(A) and NAME(C) were compared directly, then they could be found to be very different. There were a number of such cases.

To get around the erroneous links, a tightened critical level criteria was defined for the subsequent comparisons (e.g. the B and C comparison). The tighter criteria was incorporated as part of the STRICT LINK procedure. In the STRICT LINK procedure, the critical level for subsequent comparisons became one less than the length of SHORTER STRING. Final tests were run with and without the STRICT LINK.

RELATIONAL OPERATOR

After the PAIRWISE COMPARISON was executed, the aliases were grouped into sets defined by equivalent names. The relational operator procedure was carried out independently within each equivalent name set.

The procedure was done in two steps:

1. Each pair of records was tested to determine if they matched. They matched if either the CITY + STATE or ZIPCODE fields were identical for both records. These matches were defined to be the DIRECT MATCHES.
2. The transitive property (if A "matches" B and B "matches" C then it is defined that A "matches" C) was applied to extend the DIRECT MATCH relation into an equivalence relationship. The final exporter groups

were formed by the equivalence classes defined by the extended relational operator.

The following example illustrates the entire procedure and its main algorithm.

Example

Consider the following alias records of an equivalent name set.

RECORD	CITY + STATE	ZIPCODE
A	$(C+S)_1$	Z_1
B	$(C+S)_2$	Z_1
C	$(C+S)_3$	Z_2
D	$(C+S)_2$	Z_2
E	$(C+S)_4$	Z_3

1. Determine all direct MATCHES. The DIRECT MATCHES will be denoted by relation '<———>'.

Example DIRECT MATCHES

A <————> B
B <————> D
C <————> D

2. Determine all relations induced by the transitive property. This is done by the following algorithm.

ALGORITHM

i. A matrix is set up which depicts the DIRECT MATCH information. Since '<———>' is symmetric, only elements on or below the diagonal contain the MATCH information. The matrix elements e_{ij} are defined:

$$e_{ij} = 1 \quad \text{when } j <\!\!\longrightarrow i,$$
$$\quad\; = 0 \quad \text{otherwise}$$

Note: $e_{ii} = 1$, since i<———>i always.

Example-Initial Matrix

		A	B	C	D	E	(j)
(i)	A	1	0	0	0	0	
	B	1	1	0	0	0	
	C	0	0	1	0	0	
	D	0	1	1	1	0	
	E	0	0	0	0	1	

ii. The matrix is then "raked" upward to extend the direct matches. Starting with the bottom row, each row is compared to all of its above rows. When two rows are found to have at least one column in which both row elements have a value of 1 then:

1) Transfer all "one" elements from the lower row to the corresponding columns of the upper row.
2) "Zero" out the lower row.
3) Beginning with the next above row, repeat the comparison procedure.

If for a given row there is no upper row that has a corresponding "one" element, then the comparison procedure is repeated with the next above row.

iii. When all rows have been compared, then the algorithm is complete and each row that has at least one nonzero element defines an equivalence class. The "one" elements designate the class members by their column positions.

Example

1)	A	B	C	D	E
A	1	0	0	0	0
B	1	1	0	0	0
C	0	0	1	0	0
D	0	1	1	1	0
E	0	0	0	0	1

After Row E comparison

2)	A	B	C	D	E
A	1	0	0	0	0
B	1	1	0	0	0
C	0	1	1	1	0
D	0	0	0	0	0
E	0	0	0	0	1

After Row D comparison

3)	A	B	C	D	E
A	1	0	0	0	0
B	1	1	1	1	0
C	0	0	0	0	0
D	0	0	0	0	0
E	0	0	0	0	1

After Row C comparison

4)	A	B	C	D	E
A	1	1	1	1	0
B	0	0	0	0	0
C	0	0	0	0	0
D	0	0	0	0	0
E	0	0	0	0	1

After Row B comparison

The resulting equivalence classes at the completion of the algorithm are:

Group 1 = (A, B, C, D)
Group 2 = (E)

V. EVALUATION

The AUTO CODING was tested by executing it on frames which had already been VISUALLY coded. The higher the agreement levels between the AUTO and VISUAL coding the better the results were judged. Although there are undoubtedly small unknown error rates associated with VISUAL CODING, it did not appear that the results were significantly affected.

We quantified the agreement measures in terms of completeness and error rates of the AUTO CODING relative to the VISUAL CODING.

There are two types of errors that are encountered during the CODING process. These are FALSE and MISSED matches. Since it was assumed that the VISUAL CODING closely approximated the ideal coding process, then the following definitions were made:

MISSED MATCHES - Matches made by VISUAL CODING that were not made by AUTO CODING.

FALSE MATCHES - Matches made by AUTO CODING that were not made by VISUAL CODING.

In order to calculate the basic evaluation parameters, we made use of the following relationships between the number of CODING groups and matches.

Consider an EXPORTER CODING group g of N_g distinct aliases. To form that group, $N_g - 1$ matches had to be determined. The total number of matches made, T, is:

$$T = \sum_g (N_g - 1) = N - G$$

where N = total number of aliases
G = number of EXPORTER groups

T_V, T_A, G_V, and G_A are analogously defined with the V and A subscripts relating to the VISUAL and AUTO CODING, respectively.

The number of MISSED MATCHES, S, is the sum of the differences between the number of matches made by the AUTO and VISUAL coding within VISUAL groups. Moreover:

$$S = \sum_v S_v$$

where $S_v = ((\sum_a \alpha_{va} X_{va}) - 1) - \sum_a \alpha_{va}(X_{va} - 1)$

with X_{va} = number of records coded as part of VISUAL GROUP v and AUTO GROUP a

and α_{va} = 1, when there exists at least one record coded as part of VISUAL GROUP v and AUTO GROUP a,
= 0, otherwise.

Note, the first term counts the matches made by the VISUAL coding and the second term counts the ones made by the AUTO CODING. Clearly, S_V reduces:

$$S_v = \sum_a \alpha_{va} - 1 \qquad \text{and}$$

$$S = \sum_v \sum_a \alpha_{va} - G_v \qquad \text{equation (1)}$$

In a parallel fashion, the number of FALSE MATCHES, F, is:

$$F = \sum_a \sum_v \alpha_{va} - G_A \qquad \text{equation (2)}$$

From equations (1) and (2):

$$G_A + F = G_V + S \text{ and}$$

$$N + G_A + F = N + G_V + S \qquad \text{equation (3)}$$

Since
$T_V = N - G_V$ and $T_A = N - G_A$ then from equation (3) the following holds:
$T_V = T_A - F + S$

With the assumption that the T_V is nearly the total number of valid matches possible, then the completeness measure of the AUTO CODING is defined:

$$\frac{T_A - F}{T_V} \times 100$$

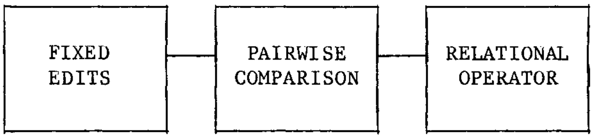
The natural definition for the FALSE MATCH error rate is:

$$\frac{F}{T_A} \times 100$$

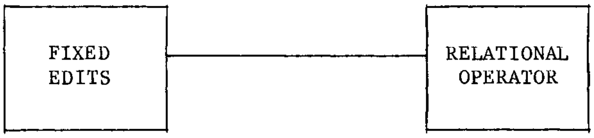
The basic methodology was altered in order to look at the impact of the PAIRWISE COMPARISON and LINKING criteria. Three slightly differing methods were each run on two separate data sets, denoted as FRAME 1 and FRAME 2. The method factors are presented below:

METHOD FACTORS

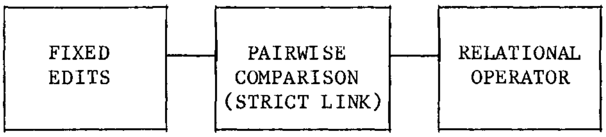
BASIC METHODOLOGY:



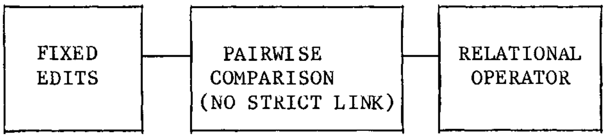
METHOD A (MINUS PAIRWISE COMPARISON)



METHOD B (STRICT LINK)



METHOD C (NO STRICT LINK)



COMPLETENESS

METHOD	$\frac{T_A - F}{T_V} \times 100$	
	FRAME 1	FRAME 2
A	65.5	64.5
B	82.5	83.2
C	86.6	86.7

ERROR RATE - FALSE MATCHES

METHOD	$\frac{F}{T_A} \times 100$	
	FRAME 1	FRAME 2
A	2.1	2.1
B	5.0	3.6
C	8.6	6.9

The three methods A, B, and C were designed so that individual effects of the PAIRWISE COMPARISON and the STRICT LINK could be analyzed. Note, all matches identified in Method A were found in Method B and all matches found in Method B were also identified in Method C.

The results were consistent for both data sets. Methods B and C achieved a completeness factor of over 80%. These were most encouraging developments.

From the COMPLETENESS data, it is clear that the FIXED EDIT and RELATIONAL OPERATOR functions accounted for the majority of the valid matches that were identified. The PAIRWISE COMPARISON (with the STRICT LINK) proved quite effective by boosting the COMPLETENESS FACTOR an average of 18%.

The ERROR data revealed that higher COMPLETENESS factors coincided with increased FALSE MATCH rates. Consider the relative error rates of the set of additional matches as defined below:

Relative Error Rate

$$= \frac{F(A)}{T_A(A)} \quad \text{for Method A}$$

$$= \frac{F(B) - F(A)}{T_A(B) - T_A(A)} \quad \text{for Method B}$$

$$= \frac{F(C) - F(B)}{T_A(C) - T_A(B)} \quad \text{for Method C}$$

The table below shows that for the set of additional matches made by Method C nearly half of those were found to be false. The STRICT LINK incorporated as part of Method B reduced the FALSE MATCH rates without significantly lowering the COMPLETENESS factors.

RELATIVE ERROR RATES FOR ADDITIONAL MATCH SET

Method	Relative Error Rate (%)	
	FRAME 1	FRAME 2
A	2.1	2.1
B	14.7	8.6
C	48.1	48.6

VI. PROGRAMMING FACTORS

The AUTO CODING software was relatively inexpensive to run. The computer code was not excessively complicated and was written in PL-1 and SAS. Although our experience dictated the choice of languages, it should be noted that the PAIRWISE COMPARISON and FIXED EDIT processes depend greatly on the handling of character strings. This capability is a strong point of PL-1.

VII. IMPLEMENTATION AND FUTURE WORK

Three important considerations for the implementation of the AUTO CODING software are:

1. The adequacy of the input alias data in terms of uniform formats and transcription errors.

2. The development of CORRECTION and VERIFICATION procedures which maximize the efficiency of the overall CODING process.

3. The capture of historical CODING data for research and development purposes.

The success of our software was due to the relatively good quality of the input data. Therefore, we will concentrate our effort in other areas.

The development of efficient CORRECTION and VERIFICATION procedures is key to the maximization of the time and cost savings we hope to achieve. It remains imperative that the final outputs of the overall CODING process be as accurate as possible. Therefore, the gains to be made by the implementation of the AUTO CODING must be ensured by the development of efficient quality control procedures. We have much work still ahead in this area.

Ironically, the errors committed by the AUTO CODING should be its greatest source of improvements. We plan to have the capability of capturing the ERROR information for an ongoing analysis with a growing pool of data. In the near future, we hope to use this data for:

1. the development of a RISK model that depicts the relationship between the PAIRWISE COMPARISON factors and FALSE MATCHES. This model would be used to fine-tune the CRITICAL LEVEL criteria.

2. the development of additional FIXED EDIT RULES.

References

[1] Handbook of Methods, United States Department of Labor, Bureau of Labor Statistics, Bulletin 2134-1, 1982

[2] Carpenter, J. F., Bishop, T. R., and Goudie, G. S., System for Matching Company Documents, 1978 Survey Research Methods Section Proceedings of the American Statistical Association, pp.505-508

Acknowledgment

The authors thank Gwyn Ferguson and Catreeda Lloyd for their careful review and preparation of the typed manuscript.