A system to monitor the quality of automated coding of textual answers to open questions

Stefania Macchia[*] and Marcello D'Orazio[**]

Italian National Statistical Institute (ISTAT),
Methodological Studies Department
Via C. Balbo, 16. 00184, Rome – ITALY

Keywords: ACTR; Coding Precision; Labour Force Survey; Stratified sampling.

Abstract

The Italian National Statistical Institute (ISTAT) carried out some tests of automated coding of textual answers regarding Occupation, Education level, Industry, etc, using the ACTR system (Automated Coding by Text Recognition). The good results obtained led ISTAT to perform a further analysis of a large sample of textual data, in order to define a standardised procedure for reference when using ACTR instead of manual coding during a survey. The analysis shown in this paper aims at building up a system to monitor the quality of the results of automated coding and at verifying the improvements which can be achieved using the results of the monitoring activity to integrate the automated coding environment.

1. Introduction

Coding written-in answers for open questions of statistical surveys typically requires dealing with the problem of their variability, depending on the cultural background of the respondents, on their ways of speaking and finally on how interested they are in co-operating. The written answers are often generic or ambiguous, since respondents are not expert in the classifications and so they respond without thinking that their answers have to be coded. The same may happen with interviewers (even if trained in advance) who are not always used to obtaining answers suitable to be easily coded.

Automated coding can help in solving the specific problems of costs, time and quality connected with the coding activity. In fact, manual coding implies high costs due to hiring, training and supervising coding personnel. It requires a long time, especially for complex questions such as Industry or Occupation, and an attempt to reduce time can negatively affect the quality of results. Finally, manual coding does not ensure any standardisation of the process (it is not sure that two different people assign the same correct code to the same textual description). The process is strongly influenced by factors related to knowledge of the classifications, skill and conscientiousness of coding clerks.

[*] E-mail: macchia@istat.it; tel. +39 06 4673 2157; fax +39 06 4788 8069.
[**] E-mail: madorazi@istat.it; tel. +39 06 4673 2278; fax +39 06 4788 8069.

For these reasons various countries, for instance France, the United States, Canada and the United Kingdom have developed and are successfully using automated coding systems. In France, Lorigny (1988) developed QUID (QUestionnaires d’IDentification), which was used in a number of socio-economic surveys. Later, the SICORE system (*Système Informatique de COdage des Réponses aux Enquêtes*) was designed to code different variables (Rivière, 1994).

In the United States, automated coding has been deeply studied and investigated. First papers appeared in the 1970's (O’Reagan, 1972; Corbett, 1972); other interesting papers are those of Hellerman (1982) and Appel and Hellermann (1983). For the 1990 Census, the U.S. Bureau of the Census developed the Automated Industry and Occupation Coding System (AIOCS); this system was adopted in the Current Population Survey and in the Survey of Income and Program Participation too (Lyberg and Dean, 1992). During the 1990's, the Bureau of the Census continued research by considering other possible techniques and systems for automated coding of Industry and Occupation (Creecy et al., 1990 and 1992; Gillman and Appel, 1994 and 1999). Still in the U.S., the Center for Health Statistics developed CLIO (Classification of Industry and Occupation), a system derived directly from the one used to code cause of death (Harris and Chamblee, 1994). Research by Statistics Canada led to release of the ACTR (Automated Coding by Text Recognition) system, which went into production in 1986 (Wenzowski, 1988). Actually, an updated release of ACTR is used to code different variables for the Census of Population and the Labour Force survey. The United Kingdom currently uses PDC (Precision Data Coder), which is a language-specific software, initially designed for Industry coding. However, to code the different textual variables observed in the (UK’s) current Census of Population, it was decided to adopt a more generalised software program such as ACTR.

Considering all these experiences, in 1998 ISTAT decided to test an automated coding system. Instead of developing new software it was decided to use the third release of the ACTR systemsupplied by Statistics Canada. ACTR was chosen since it was language-independent and seemed easily adaptable to the Italian language, unlike other systems that were language-specific, such as for instance PDC. Moreover, as already mentioned, ACTR is a generalised system, so it can be used for more than one coding application. In addition, it has already been successfully used by other National Statistics Institutes (Tourigny and Moloney, 1995).

2. The ACTR system

ACTR's philosophy is based on methods originally developed at the U.S. Bureau of the Census (Hellerman, 1982), but uses matching algorithms developed at Statistics Canada (Wenzowski, 1988). The coding activity follows a quite complex phase of text standardisation, called *parsing*, that provides fourteen different functions such as character mapping, deletion of trivial words, definition of synonyms, removal of suffixes (these functions are completely managed by the users). The *parsing* aims to remove grammatical or syntactical differences so as to make equal two different descriptions with the same semantic content. The parsed answer to be coded is then compared with the parsed descriptions of the dictionary, the so-called *reference file*. If this search returns a perfect match, called *direct match*, a unique code is assigned, otherwise the software uses an algorithm to find the best suitable partial (or fuzzy) matches, giving an *indirect match*. In practice, in the latter case the software takes out of the reference file all the descriptions that have at least one parsed word in common with the one given by the respondent and assigns them a score. This score, standardised between 0 and 10 (10 corresponds to a perfect match), is computed as a function of the weight given to each single word in common, which is in inverse relation to its frequency of occurrence in the dictionary.

The system orders by decreasing score ($S_1 \geq S_2 \geq \ldots \geq S_n$) the descriptions selected from the *reference file* and compares them with three user-defined thresholds: the lower limit ($S_{\min}$), the upper limit ($S_{\max}$) and the minimum score difference (ΔS). If $S_1 \geq S_{\max}$ and $\left(S_1 - S_2\right) \geq \Delta S$ the description with the score S_1 is said to be a *unique* winner and a unique code is assigned to it. If the first two (or more) descriptions are greater or equal to $S_{\max}$ ($S_1 \geq S_{\max}$ and $S_2 \geq S_{\max}$) but their difference is less than the minimum score difference ($S_1 - S_2 < \Delta S$), the system returns both as winners (*multiple* winners).The same happens if $S_{\min} \leq S_2 \leq S_1 \leq S_{\max}$;notice that in this case the similarity between the description to be coded and those selected from the *reference file* is lower than in the previous case. Finally, there are no winners if all the scores are less than $S_{\min}$ ($S_1 < S_{\min}$) and the system returns a *failed* message.

For *unique* winners no human intervention is required, while all the other cases need to be evaluated by expert coders to choose which of *multiples* will be the right one or whether to code at all the *failed* matches.

The following example, concerning Occupation, clarifies how the indirect match works. The description "*esercente di art. di abbigliamento di vario genere (esclusi i pellami)*" ["trader of clothes art. of various kinds (with exception of leather)"], after the parsing process we defined, becomes "*abbigliament commerciant*" ["clothes dealer", suffixes removed] and matches with the sentence of the reference file (actually used): "*esercente di negozio di abbigliamento*" ["shop trader of clothes"]. In practice, the parsing first operates on strings, eliminating certain clauses ("*esclusi i pellami*"), deleting non-informative *strings* ("*di vario genere*"), replacing strings with synonyms and so on. It then operates on words, replacing words with synonymous ("*esercente*" becomes "*commerciante*"), deleting non-informative words ("*di*", "*i*") and removing suffixes from all words that do not have to be treated as exceptions. As the two sentences are similar but not identical, there is an indirect match with a score of 9.33; this score is greater than the threshold $S_{\max} = 8.0$ and, given that $\left(S_1 - S_2\right) > \Delta S = 0.2$, a unique code is assigned to the starting description.

Unfortunately, the indirect matching mechanism can produce errors. For example consider the description "*addetto ai servizi ausiliari*" ["assigned to auxiliary services"]; it would match (with the actual reference file) with "*addetto ai servizi ausiliari del reattore*" [“assigned to auxiliary services of the reactor”] and, having a high score, it would be uniquely coded. But, as it can be seen, the original description does not refer to any reactor, instead it should match with the code corresponding to the description “*personale inserviente negli uffici*” [“office attendant”].

Hence, when an automatic coding system is in production, the quality of its results has to be monitored and coding errors have to be used to update the application environment so as to prevent further errors of the same kind.

3. The construction of the automatic coding environment

Using ACTR requires a phase of *training*, which involves building the environment of the coding system. The first step of the training requires the construction of coding dictionaries (lists of texts with the corresponding codes); afterwards the system has to be adapted to the language and to each classification; and finally it has to be tested.

The building of coding dictionaries (*reference files*) is the heaviest activity, as their quality and their size deeply affects the performance of automated coding. Basically, it involves: (i) re-elaborating the textual descriptions used in classification manuals in order to make them simple, analytical and unambiguous; and (ii) integrating the classification dictionaries with information based on expert knowledge, with descriptions coming from other related official classifications and with empirical response patterns taken from previous surveys (in order to reproduce the respondents' natural language as close as possible).

The already-mentioned parsing functions, which are managed through *parsing files*, allow users to adapt the system to the language and to the classification. The implementation of these *parsing files* is very easy and does not require the user to be a computer expert.

As far as the adaptation to Italian is concerned, in all the applications we built, we decided to define as irrelevant the articles, conjunctions and prepositions, and we removed suffixes which determine singular and plural. Only in considering Occupations was it necessary to remove the gender suffixes too. On the other hand, the definition of synonyms, both at string and at word level, is a job that requires more effort, since the classification is complex and answers can vary in their “wording”. In order to clarify this aspect with figures, well over 2,000 synonyms were necessary when Occupation type was considered, whereas just 287 have been defined for Education Level.

Up to now, we have trained the system to work with three variables: Occupation, Industry and Education Level. Each variable shows a different level of complexity, due to the corresponding classification complexity and to the expected variability in the “wording” of answers (both these aspects influence the results of automated coding, as confirmed by the experiences in other countries).

The benchmark files we used to train the system for the three mentioned variables were a sample of 9,000 households drawn to perform a Quality Survey on 1991 Population Census and a sample drawn from the Intermediate Census of Industry and Services ("Short Form survey"). To train ACTR we ran it repeatedly on these samples, selecting each time the empirical answers to be added to the dictionaries and at the same time, improving the parsing process until the highest possible number of correct unique matches was reached. The rates of matching (answer phrase: single code) obtained at the end of the runs were: 72.5% for Occupation; 86.6% for Education Level; 54.5% and 73.0% respectively for Industry on the first and second sample (this difference is due to households' difficulty in answering this question). Hence they were in line with the results obtained by other Countries (Lyberg and Dean, 1992).

4. First results of automated coding

After training the system, it needs to be tested in order to verify if the application environment, built using small samples, is suitable to be used for data-sets of bigger size. For this purpose the quality of automated coding has to be measured in terms of *recall*, i.e. the percentage of codes automatically assigned, as well as in terms of *precision*, i.e. the percentage of correct codes automatically assigned.

Table 1 shows the results obtained in terms of *recall* on data collected in the 1994 Health Survey, the 1998 Labour Force survey (four quarters collected and already manually coded), the 1999 Labour Force pilot survey and the 1998 Intermediate Census of Industry and Services ("Long Form survey"). These results are consistent with those obtained during the system training.

Source of texts	Occupation		Industry	
	No. Texts	Recall	No. Texts	Recall
Health Survey	33,735	72.3%	–	–
1998 Labour Force Survey	356,231	71.9%	–	–
1999 Labour Force Pilot Survey	1,307	67.6%	1,252	44.6%
"Long Form Survey"	–	–	37,161	63.0%

Table 1 – Some results on recall of automatic coding

As far as *precision* is concerned, with the aid of expert coders who analysed all the automatically assigned codes, it was possible to achieve the results shown in the following table.

Source of texts	Occupation		Industry	
	Uniquely Coded	Precision	Uniquely Coded	Precision
Health Survey	24,404	97.0%	–	–
1999 Labour Force Pilot Survey	884	99.0%	558	86.0%

Table 2 – Precision of automatic coding[1]

It was not possible to do the same thing in the Labour Force survey, due to its great amount of texts (256,748 texts coded as *unique*); here the *precision* can be evaluated only on a sample basis. Hence the need to build a system to monitor the quality of automatic coding, which can determine the extraction of sample of texts that have to be submitted to expert coders (section 5.).

5. Monitoring and enhancing the quality of automatic coding of great amount of texts

We analysed the textual answers for the 1998 Labour Force survey (four quarters collected) with the purpose of: (i) thoroughly evaluating the performance of the automatic coding; (ii) building up a quality monitoring system; and (iii) doing further training of the coding environment, whose main purpose is to enrich the dictionary with new texts.

As a first step we quantified how many "different" texts existed in the original file and defined some frequency classes, so as to evaluate the performance of the system, class by class. To identify the "different" texts, we performed a kind of "raw standardisation" with only a few parsing functions, so as to delete from descriptions the articles, conjunctions, prepositions and suffixes (in practice all the elements that determine the gender of words, the singular/plural, etc.). As can be seen in Table 3, the initial 356,231 texts were reduced to 59,562 different ways of describing the occupation. On the other hand, 74% of these descriptions occurred only once in the original file, thus proving a high variance in wording of answers, if compared with the 6,319 official elementary definitions derived from just 599 occupations listed in the classification manual.

Original Texts	"Different" Texts	Occurrence					
		1	2	3–10	11–50	51–1,000	1,001–10,000
356,207	59,562 (100.00%)	43,349 (73.78%)	7,344 (12.33%)	6,404 (10.75%)	1,783 (2.99%)	640 (1.07%)	41 (0.07%)

Table 3 – Distribution of "different" texts by classes of occurrence

[1] Evaluation of *precision* for Long Form survey is still in progress.

5.1. Evaluating the performance of automatic coding environment

A primary indicator of the performance of the automatic coding environment is achieved by comparing its *recall* on the original data-set (the one with all nonparsed texts) with that of "different" texts. Obviously the system *recall* on this latter file is lower, as can be seen in table below.

ACTR output	*Recall*	
	N. texts	%
Unique	19,404	32.5
Multiple	20,537	34.5
Failed	19,620	33.0
Total	59,561	100.0

Table 4 – ACTR results on "different" texts: recall

Recall grows as frequency class becomes higher (Table 5). In particular, for "different" texts occurring only once, ACTR assigned a unique code in 27% of cases, while for texts occurring more than 100 times, this rate goes beyond 79%. This means that the actual reference file already includes most of the occupation descriptions that occur frequently in common speaking.

ACTR output	Occurrence									
	1		2–10		11–100		101–1,000		1,001–10,000	
	No.	%	No.	%	No.	%	No.	%	No.	%
Unique	11,786	27.2	5,869	42.7	1,437	69.0	273	79.6	39	95.1
Multiple	15,735	36.3	4,303	31.3	431	20.8	66	19.2	2	4.9
Failed	15,828	36.5	3,576	26.0	212	10.2	4	1.2	0	0.0
Total	43,349	100.0	13,748	100.0	2,080	100.0	343	100.0	41	100.0

Table 5 – ACTR results on frequency classes of "different" texts: recall

5.2. Lack of standardisation of manual coding process

The quality of automated coding can be further evaluated by comparing it with the level of standardisation in the manual coding process.

As the Labour Force data were previously manually coded, we could quantify the different codes assigned by manual coders to the same text (Table 6).

Texts frequency	Different codes assigned to “equal” texts			
classes	*Max N.*	*Mean*	*Median*	*Mode*
2	2	1.27	2	1
3–5	5	1.84	3	1
6–10	10	2.68	3	1
11–50	33	4.65	4	2
51–100	42	10.05	8	4
101–1,000	119	18.65	14	7
1,001–10,000	389	67.46	51	33

Table 6 – Lack of standardisation in manual coding

The results in this table show how low the level of standardisation of manual coding is. The discrepancy between codes assigned by different operators can usually be ascribed to different interpretations of the response text, to different knowledge of the classification and to misunderstandings. On the other hand, there is surely a percentage of texts (which we could not quantify) to which operators assigned different codes in view of some other information taken from other correlated questions in the questionnaire (for instance Industry).

5.3. The system to monitor the quality

Given the characteristics of ACTR, the sample of n “different” texts to be checked has to be drawn from those uniquely coded with a score less than 10 ($N = 13{,}821$). In fact a text coded with a score of 10, corresponding to a direct match, has a correct code (unless there are some mistakes in the *reference file*).

We decided to use a stratified random sampling design to draw the sample. In practice, texts were first stratified according to their frequency of occurrence M_j; then, within each stratum a simple random sample (without replacement) of texts was selected. The strata coincided with the previously defined classes of occurrences with exception of the “1,001–10,000” one, given that all its 41 “different” texts had a coding score equal to 10, i.e. they were all correctly coded.

In deciding the sample size it is possible to choose between two different strategies: (i) to compute the overall sample size and then allocate it between the strata; or (ii) to compute the sample size independently for each stratum, according to the precision of estimates required in each of them.

With the first strategy the overall optimal sample size can be approximately computed by using *Neyman allocation* (see e.g. Cochran, 1977, p. 105). In this circumstance it is important to decide *a priori* how the sample should be allocated between strata. For example, with proportional allocation, the sample is allocated according to the relative size of each stratum $W_h = N_h / N$. However, with the problem at hand, a better approach could be that of allocating the sample according to the relative sum of frequencies of "different" texts in the same class, so as to sample more "different" texts with higher importance.

The advantage of this procedure is that of computing directly the overall sample size, given an allocation criterion. The disadvantage is that for some stratum the optimal sample size may be greater then the entire stratum size (N_h); here, one has to revise the allocation following Cochran (1977, p. 104).

The alternative strategy avoids this last problem; it involves deciding the optimal sample size independently from stratum to stratum using each time the following expression (see Cochran, 1997, pp. 75-76):

$$n_h^* = \frac{n_{0h}}{1 + (n_{0h} - 1)/N_h}$$

with

$$n_{0h} = \frac{z_{1-\alpha/2}^2 \tilde{\pi}_h (1 - \tilde{\pi}_h)}{d_h^2}, \qquad h = 1,2,\ldots,L .$$

In this expression $\tilde{\pi}_h$ is the hypothesised *precision* of automated coding for texts belonging to class *h*; d_h represents the overall margin of error allowed in estimating the unknown *precision,* π_h, of automated coding and *z* is the percentile of standardised normal distribution such that $\Pr(|\hat{\pi}_h - \pi_h| \geq d_h) = \alpha$.

Then, the overall sample size is achieved by summing up the so obtained optimal sample sizes: $n = n_1^* + n_2^* + \ldots + n_L^*$. The problem with this procedure is that *n* may easily explode if some n_h^* values are too large.

We used this latter strategy in deciding the size of the sample of text to submit to expert coders. An equal *precision* rate of automated coding in each class, $\tilde{\pi}_h = 0.75$ was hypothesised, while the margin of error *d* was progressively reduced (4th column of the Table below) in higher classes of occurrences of "different" texts; this guaranteed estimates with a higher *precision* for heaviest "different" texts. The approximate sample size computed for various classes with $\alpha = 0.05$ ($z_{0.975} = 1.96$) can be found in the 4th result column.

Classes of occurrences	Number of different texts (N_h)	Hypothesised precision of autom. coding ($\tilde{\pi}_h$)	Margin of error (d_h)	Approximate optimal sample size (n_h^*)	Sampling fraction ($f_h = n_h^*/N_h$)
1	10,007	75.0%	±5.0%	148	1.48%
2	1,756	75.0%	±5.0%	138	7.86%
3–5	1,187	75.0%	±4.5%	160	13.48%
6–10	473	75.0%	±3.0%	222	46.93%
11–50	349	75.0%	±2.5%	221	63.32%
51–100	33	75.0%	±1.0%	33	100.00%
101–1,000	16	75.0%	±1.0%	16	100.00%
Tot.	13,821			938	6.79%

Table 7 – Optimal sample sizes in the strata

The sample of 938 texts was then submitted to expert coders, in order to evaluate if ACTR had assigned correct codes. In this way it was possible to estimate *precision* for each class of occurrences and hence for all the 13,821 "different" texts. The estimates, computed using the theory of stratified random sampling (cf. Cochran, 1977, pp. 90-96), can be found in Table 8, with the corresponding values useful to derive the 95%-confidence interval (last column of the table).

As can be seen, we estimated that 75.77% of the 13,821 "different" texts were correctly coded by ACTR. True *precision* lies between 70.58% ($= 75.77 - 5.19$) and 80.95% ($= 75.77 + 5.19$) approximately with a probability of 0.95. The *precision* tends to be higher (over the 80%) for the last classes. Notice that for the last two classes we do not have an estimate but the true *precision*, as all texts (rather than a sample) were checked. Here the coding *precision* is over the 80% and this further proves that the system works well with more frequent descriptions.

Classes of occurrences	"Different" texts	Sample size	Sampling fraction (%)	Estimated precision (%)	Estimated margin of error
1	10,007	148	1.48	74.32	±6.99
2	1,756	138	7.86	81.88	±6.17
3–5	1,187	160	13.48	78.13	±5.96
6–10	473	222	46.93	73.42	±4.23
11–50	349	221	63.32	80.09	±3.19
51–100	33	33	100.00	87.88	–
101–1,000	16	16	100.00	81.25	–
Tot.	13,821	938	6.79	75.77	±5.19

Table 8 – Estimated precision of automatic coding of different texts

If we consider also the 6,083 ($=19{,}904-13{,}821$) “different” texts coded with a score of 10 (all correctly coded), the overall estimated *precision* goes up to 83.17% of 19,904 “different” texts.

The estimated *precision* of automated coding when applied to original texts can be easily derived from that of the “different” texts, by considering the occurrences of these latter ones (Table 9). In practice each “different” text can be viewed as a cluster of original texts and the theory of cluster sampling allows us to derive the estimates of *precision* and the corresponding 95%-confidence intervals reported in Table below.

Classes of occurrences	“Different” texts	Original Texts	Estimated precision (%)	Estimated margin of error
1	10,007	10,007	74.32	±7.01
2	1,756	3,512	81.88	±6.19
3-5	1,187	4,337	78.34	±6.55
6-10	473	3,492	73.40	±4.52
11-50	349	7,320	86.29	±5.08
51-100	33	2,214	87.49	–
101-1,000	16	3,731	81,96	–
Tot.	13,821	34,613	79.70	±2.57

Table 9 – Estimated precision of automatic coding of original texts

It is estimated that the 79,7% (27,586 texts) of the 34,613 original texts uniquely coded with a score less than 10 were coded correctly. The true *precision* lies between 77.13% ($=79.7-2.57$) and 82.26% ($=79.7+2.57$), with an approximate confidence of 0.95. Here too, if we consider the 222,135 original texts uniquely coded with a score equal to 10, it comes out that 249,721 of the 256,748 original texts uniquely coded had a correct code (i.e. 97.26%). This last estimate is in line with the one obtained for the Health survey (see Table 2).

Thus, with a small but well designed sample (in this case 6.79% of single texts) it was possible to evaluate the precision of automated coding results with a high confidence.

5.4. First results of the further training of coding environment

The further training phase consists in adding new texts to the dictionary and updating the coding environment: (i) to prevent further texts being processed as coding errors found ; and (ii) to increase the future *recall* rates.

To prevent further coding errors, the sample of “different texts” for which ACTR did not assign a correct code, as determined byexpert coders, needs to be analysed.

In order to increase the future *recall* rate, texts for which ACTR was not successful in assigning a single code need to be examined, including coded texts having enough informative content to be assigned a unique code (i.e. those which are not too generic, or which describe concepts which can be directly linked with single codes). In this regard, it is convenient to examine first the more frequent ones, while the analysis of texts belonging to lower frequency classes, given their minor importance, can be restricted to only a sample.

We analysed the failed matches returned by ACTR when coding the file of "different" texts occurring more than 10 times. By analysing only 216 different texts (212 belonging to the "11–100" class of occurrences and 4 to the "101–1,000" one) we added 299 new texts to the reference file and 46 synonyms (at both the level of string and of the word).

The *recall* rate obtained on the original text data-set after this further training activity are shown in the following table.

ACTR output	Recall	
	No. Texts	%
Unique	269,485	75.6
Multiple	58,848	16.6
Failed	27,898	7.8
Total	356,231	100.0

Table 10 – ACTR results on original Labour Force survey sample after the further training: recall

As can be seen, the percentage rises from 71.9% to 75.6% and is likely to be even higher if we had also analysed the multiple matches.

Finally, we verified if the update of the coding environment for Occupation, achieved by analysing Labour Force descriptions, could imply better results for other coding applications performed on data from other surveys. For this purpose, we automatically coded again the Health survey texts, as it was the next biggest file we had at our disposal, after the Labour Force one. As shown in table 11, the *recall* rate grows from 72.3% to 75.1%, thus confirming that the outcomes of each coding application represent a precious feed-back to update the coding environment and give the chance of achieving higher *recall* rates.

ACTR output	Recall	
	No. Texts	%
Unique	25,337	75.1
Multiple	5,827	17.3
Failed	2,571	7.6
Total	33,735	100.0

Table 11 – ACTR results on Health survey sample after further training: recall

6. Conclusions

As mentioned in the previous sections, ISTAT spent much work and time in order to introduce automatic coding of written-inanswers to open questions regarding Occupation, Education Level and Industry by means of the ACTR system. Most of the work involved building the *reference files* and the corresponding *parsing files* for both Occupation and Industry. The first results obtained in this direction (section 4.) were encouraging, especially if compared with those of manual coding, and led us to further improve the automatic coding environment, using all available sources of textual descriptions to enhance the *reference files* and to refine the *parsing* step. Alongside this activity, we thought it was necessary to introduce an evaluation procedure so as to quantify the quality of ACTR output (section 5.). This procedure was kept as general as possible in order to get a reliable idea, even if on a sample basis, of how well ACTR codes texts that do not exactly correspond to descriptions of the *reference file*. We performed this evaluation step on a large amount of texts regarding Occupation (section 5.3) and the results obtained were particularly satisfactory (overall coding precision was estimated to be about 97%).

All the work invested in ACTR training and the good results obtained in the testing/evaluation phase convinced us that it can successfully be adopted for use in different surveys, even to code such complex descriptions as the Occupation and Industry ones, giving more consistent results than those of manual coders. Moreover, these results seem to be achievable at a lower cost; gains are likely to increase with the amount of descriptions collected. In any case, the application of ACTR should constantly be monitored in all its phases.

Despite the advantages, it has to be kept in mind that the application of ACTR still presents a problem in cases where the system fails in assigning a unique code. Different solutions are available. One, for example, could be that of trying to code by making use of additional information derived from related questions of the same form. This could be achieved automatically or with the intervention of expert coders, maybe aided by an assisted coding system. If this does not work, and if the number of unsolved cases is not high, it may be necessary to consider re-contacting the respondents. Therefore further investigation is needed in order to choose the strategy that performs better in terms of the number of cases solved, costs and quality of final results.

7. References

[1] Appel, M. and Hellerman, E., 'Census Bureau experience with Automated Industry and Occupation Coding', "Proceedings of Section on Survey Research Methods", *American Statistical Association*, 1983, pp. 32-40.

[2] Chen, B., Creecy, R. and Appel, M., 'Error control of automated industry and occupation coding', *Journal of Official Statistics*, Vol. 9, 1993, pp. 729-745.

[3] Cochran, W. G., *Sampling Techniques,* 3rd edition, Wiley, New York, 1977.

[4] Corbett, J.P., 'Encoding from free word descriptions', *Unpublished manuscript,* U.S. Bureau of the Census, 1972.

[5] Creecy, R. H., Causey, B. D., and Appel, M. V., 'A Bayesian classification approach to automated industry and occupation coding', *Paper presented at the American Statistical Associations Joint Statistical Meetings*, Anaheim, CA, 1990.

[6] Creecy, R. H., Masand, B. M., Smith, S. J. and Waltz, D. L., 'Trading MIPS and memory for knowledge engineering', *Communications of the ACM*, Vol. 35, 1992, pp. 48-68.

[7] De Angelis, R. and Macchia, S., 'Applying automated coding to the pilot survey of next population census: a challange', *Paper presented at Conference on New Techniques and Technologies for Statistics,* Sorrento, Italy 4-6 November 1998, pp 309-314.

[8] Dumicic, S. and Dumicic, K., 'Optical reading and automatic coding in the Census '91 in Croatia', *Paper presented at Conference of European Statisticians, Work Session on Statistical Data Editing*, Cork, Ireland 17–20 October 1994.

[9] Gillman, D. and Appel, M. V., 'Automated coding Research at the Census Bureau'. *Research Report*, N. 4, US Bureau of the Census, Washington, 1994.

[10] Gillman, D. and Appel, M. V., 'Developing an automated industry and occupation coding system for the 2000 census'. *Paper presented at Joint Statistical Meeting* held in Baltimore, 1999.

[11] Harris, K. W. And Chamblee, R. F., 'Evaluation of an automated industry and occupation coding system', *Paper presented at Joint Statistical Meeting* held in Toronto, 1994.

[12] Hellermann, E., 'Overview of the Hellerman I&O Coding System'. *US Bureau of the Census internal paper*, Washington, 1982.

[13] Kalpic, D., 'Automated coding of census data', *Journal of Official Statistics*, Vol. 10, 1994, pp. 449-463.

[14] Knaus, R., 'Methods and problems in coding natural language survey data', *Journal of Official Statistics*, Vol. 1, 1987, pp. 45-67.

[15] Lorigny, J., 'QUID, A general automatic coding method', *Survey Methodology*, Vol. 14, 1988, pp. 289-298.

[16] Lyberg, L. and Dean, P., 'Automated Coding of Survey Responses: an international review', *Paper presented at Conference of European Statisticians*, Work session on Statistical Data Editing, Washington DC, 1992.

[17] Massingham, R., 'Data capture and Coding for the 2001 Great Britain Census', *paper presented at XIV Annual International Symposium on Methodology Issues*, 5–7 November 1997, Hull, Canada.

[18] O'Reagan R. T., 'Computer assigned codes from verbal responses', *Communications from the ACM*, Vol 15, 1972, pp. 455-459.

[19] Riviere, P., 'Le système de codification automatique SICORE', *Working paper, Conference des Statisticiens Européens,* Séminaire ISIS 1994, Bratislava

[20] Tourigny, J. Y. and Moloney, J., 'The 1991 Canadian Census of Population experience with automated coding', *Paper presented at United Nations Statistical Commission on Statistical Data Editing*, 1995.

[21] Wenzowski, M. J., 'ACTR – A Generalised Automated Coding System', *Survey Methodology*, Vol. 14, 1988, pp. 299-308.