Analysis of Textual data for integrating an automated coding environment system and building a system to monitor the quality of its results

Stefania Macchia, Marcello D'Orazio

ISTAT – Methodological Studies Department – Rome – Italy

Abstract

Actually Italian National Institute of Statistics (ISTAT) is evaluating the chance of using a software for automatic coding of textual responses to questions about occupation, education level etc.. The system chosen is ACTR (Automated Coding by Text Recognition) developed by Statistics Canada. A first test of the system was carried out with data from the quality survey on Population Census of year 1991. The good results obtained led to perform a further analysis with textual data from Labour Forces Survey. The purpose was to define a standardised procedure which to refer when ACTR is used during a survey instead of a manual coding. In particular, the analysis carried out in this paper aims at developing a procedure to integrate the basic automated coding environment and to build up a system to monitor the quality of the results of automated coding.

Keywords: Automated Coding; Quality Monitoring; Textual Data.

1. Introduction

Manual coding of responses to open questions is a time-consuming job and does not guarantee in terms of standardisation of the process. That is why in 1998 Istat decided to test an automated coding system. The software selected is ACTR (Automated Coding by Text Recognition, v. 3), a package developed by Statistics Canada. The choice fell on ACTR because it is a generalised system, independent from the language, already successfully used by other National Statistics Institutes (Tourigny et Moloney, 1995).

2. The automatic coding system ACTR

ACTR' s philosophy lies on methods originally developed at US Census Bureau (Hellerman, 1982), but uses matching algorithms developed at Statistics Canada (Wenzowski, 1988).

The coding activity follows a quite sophisticated phase of *text standardisation*, called *parsing*, that provides 14 different functions such as characters mapping, deletion of trivial words, definition of synonymous, suffixes removal, etc.. The parsing aims at removing grammatical or syntactical differences so to make equal two different descriptions with the same semantic content. The parsed response 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 answered phrase and assigns them a score, standardised between 0 and 10 (10 corresponds to a perfect match), calculated as a function of the weight given to each single common word; the weight is inversely correlated to the frequency of occurrence of the word in the dictionary. Then, the system arranges by decreasing scores the descriptions extracted from the reference file and compares them with some user-defined threshold parameters; the results might be:

- a *unique* match, if a unique code is assigned to a response phrase;
- *multiple* matches, if several possible codes are proposed;
- a *failed* match, if no matches are found.

The first case does not require a human intervention, while the other ones have to be evaluated by expert coders.

Table below (Table 1) gives an example of indirect match. As it can be seen, the description "*esercente di art. di abbigliamento di vario genere (esclusi i pellami)*" ["trader of clothes art. of various kind (with exception of leather)"], after the parsing process, becomes "*abbigliament commerciant*" ["clothes dealer"]. In practice, at first, the parsing operates on strings, eliminating certain clauses, deleting non informative strings, replacing strings with synonymous and so on; then it operates on words, removing suffixes from all the words which do not have to be treated as exceptions. At the end, the parsed original description matches with the following sentence of the reference file: "*esercente di negozio di abbigliamento*" ["shop trader of clothes"]. 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 defined threshold parameters, so a unique code is assigned.

Parsing Results

Original Text: "esercente di art. di abbigliamento di vario genere (esclusi i pellami)"

String trimming "esercente di art. di abbigliamento di vario genere (esclusi i pellami)"

Word Characters (Translation) "ESERCENTE DI ART. DI ABBIGLIAMENTO DI VARIO GENERE (ESCLUSI I PELLAMI)"

Deletion Clauses "ESERCENTE DI ART. DI ABBIGLIAMENTO DI VARIO GENERE"

Deletion Strings "ESERCENTE DI ART. DI ABBIGLIAMENTO DI"

Replacement Strings "ESERCENTE DI ART. DI ABBIGLIAMENTO DI"

Word Characters (Elimination) "ESERCENTE DI ART DI ABBIGLIAMENTO DI"

Hyphenated Words "ESERCENTE DI ART DI ABBIGLIAMENTO DI"

Illegal Words "ESERCENTE DI ART DI ABBIGLIAMENTO DI"

Replacement Words ."COMMERCIANTE ART ABBIGLIAMENTO"

Double Words "COMMERCIANTE ABBIGLIAMENTO"

Exception Words "COMMERCIANTE ABBIGLIAMENTO"

Suffixes "COMMERCIANT ABBIGLIAMENT"

Duplicate Word Removal "COMMERCIANT ABBIGLIAMENT"

Word Sorting "ABBIGLIAMENT COMMERCIANT"

Deletion-clause count = 1.

Parsed Text: "ABBIGLIAMENT COMMERCIANT"

Table 1 - Parsing activity

Unfortunately, the indirect matching mechanism can produce errors. An example is the following one: the description "*addetto ai servizi ausiliari*" ["assigned to auxiliary services"] would match with "*addetto ai servizi ausiliari del reattore*" ["assigned to auxiliary services of the reactor"] and, having a high score, ACTR would return a unique code. As it can be seen, the original description does not refer to any reactor but should be matched with the code corresponding to the description "*personale inserviente negli uffici*" ["office attendant"]. Hence, when an automatic coding system is in production, it is always needed to monitor the quality of its results; coding errors have to be used to update the application environment so to prevent further errors of the same kind.

3. The construction of the automatic coding environment

Before using ACTR, it is required to build the environment of the coding system (the so called *system training* activity) by developing the coding dictionaries (lists of texts with the corresponding codes), adapting the system to Italian language and to each classification and, at the end, by testing it. The construction of coding dictionaries (*reference file*) is the heaviest activity, as their quality and their size deeply affects the performance of automated coding. Basically, this activity consists in the following tasks:

- re-elaborating the textual descriptions used in classification manuals in order to make them simple, analytical and unambiguous;
- integrating the classification dictionaries with information based on experts knowledge and taken from classification manuals or from other related official classifications;
- integrating the classification dictionaries 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 as many *parsing files*, allow to adapt the system to the language and to the classification. Until now, we have already "trained" the system to work with three variables: Occupation, Industry and Education Level. They present a different level of complexity due to each own classification complexity and to the expected variability in responses "wording" (as confirmed by experiences made by other Countries, both these aspects influence the results of automated coding). The benchmark file used for these purposes was a sample of 9,000 households from a Quality Survey performed on 1991 Population Census.

To train ACTR we ran repeatedly it on this sample, improving the parsing process and selecting every time the empirical responses to be added to the dictionaries, until the highest possible number of correct unique matches was reached.

The rates of matching (response phrase–single code) obtained at the end of "training" were respectively: 72.5% for Occupation, 54.5% for Industry and 86.6% for Education Level; hence in line with results obtained by other Countries (Lyberg et Dean, 1992).

4. Testing the automated coding environment system: preliminary results

As far as Occupation is concerned, it was possible to test the system with data from two surveys: 1994 Health survey (33,730 texts) and 1998 Labour Force survey (356,231 texts, corresponding to 4 quarters collected and already manually coded). The quality of automated coding was measured in terms of:

- *recall*, the percentage of codes automatically assigned;
- *precision*, the percentage of correct codes automatically assigned.

As shown in Table 2, the *recall* percentages proved that the application environment was suitable to be used for data-set of bigger dimensions even if built using a small sample. As far as *precision* is concerned, making use of expert coders who analysed all the codes assigned automatically to the Health Survey texts, it was possible to show that 97% of them was correct; unfortunately in Labour Force survey, due to its great amount of texts, the *precision* can be evaluated only on sample basis; it is necessary to build a system to monitor the quality of automatic coding, which steer in selecting the sample of texts that have to be submitted to expert coders (see par. 5.4.).

ACTR Results	**Health Survey**			**Labour Force Survey**	
	Recall		*Precision*	*Recall*	
	N	%	%	N	%
Unique	24,404	72.3	97.0	256,748	72.0
Multiple	6,213	18.4	-	67,519	19.0
Failed	3,112	9.3		31,964	9.0
Total	33,735	100.0		356,231	100.0

Table 2 – Recall and precision of automatic coding of Occupation.

5. Analysis of Labour Force textual responses

The analysis of Labour Force textual responses was aimed at:

- deeply evaluating the performance of the automatic coding;
- making a further training of coding environment, whose main activity consists in the enrichment of the dictionary with new texts;
- building a quality monitoring system.

As a first step we quantified the number of "different" texts present in the original file and defined some classes of frequency, so to evaluate the performance of the system class by class.

To identify the "different" texts, we performed a kind of "raw standardisation" with only few parsing functions, so to delete from descriptions the articles, the conjunctions, the prepositions and the suffixes (in practice all the elements that determine the gender of words, the singular/plural, etc.). As it can be seen in Table 3, the initial 356,231 texts cut down to only 59,562 different ways of describing the occupation. On the other hand, the 74% of these descriptions occurred only once in the original file, thus confirming a high variance in responses wording, mostly if compared with the only 599 occupations listed in the classification manual, which correspond to 6,319 official elementary definitions.

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.

5.1. Evaluation of the performances of automatic coding environment

The primary indicator of the performance of the automatic coding environment is obtained by comparing its *recall* on the original data-set (the one with all nonparsed texts) and on the smallest one with "different" texts. Obviously the system *recall* on this latter file is lower, as it can be seen in Table 4.

ACTR Results	*Recall*	
	N	%
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 classes become higher (Table 5). In particular, for "different" texts occurring only once, ACTR assigned a unique code in the 27.2% of cases, while for texts occurring more than 100 times, this rate goes beyond the 79%. This means that the *reference file* already includes many of occupation descriptions which occur frequently in common speaking.

ACTR Results	Occurrence									
	1		2-10		11-100		101-1,000		1,001-10,000	
	N	%	N	%	N	%	N	%	N	%
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 of the manual coding process.

As the Labour Force data were previously manually coded, we could quantify how many different codes were assigned by manual coders to the same text. The results in Table 6 show that the level of standardisation of manual coding is low. The discrepancy between codes assigned by different operators is usually to be ascribed to different interpretation of the response text, different knowledge of the classification and to misunderstandings. On the other hand, there surely is a percentage of texts (we could not quantify) to which operators assigned different codes in view of some other information taken from other correlated questions of the questionnaire (for instance Industry).

Texts frequency classes	Different codes assigned to "equal" texts			
	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 of manual coding.

5.3. The further training of coding environment

The further training phase of the coding environment consists in submitting the texts to which the system did not succeed in assigning a code to expert coders. The purpose is not only that of coding them, in fact they can be used as new texts to be added in the dictionary or to update the coding system (for instance adding synonymous or modifying the parsing files that are to be used in a further text processing).

Results shown in Table 4 and 5 are very useful to plan this activity. In order to increase the total *recall* rate, it is suggested to work at first on more frequent texts and to include in the coding environment all of them with an informative content which is exhaustive to assign a unique code (i.e. those which are not too generic or do not describe concepts which can not be directly linked with single codes). On the contrary, the analysis of texts belonging to lower frequency classes, given their minor importance, can be restricted to only a sample of them.

5.4. The system to monitor the quality

When an automatic coding system is put in the survey flow, it is necessary to monitor constantly its performances in terms of *precision*. Unfortunately, it will never be possible to check all the coded texts but only a small sample of them. The texts to check are the one uniquely coded but with a score less than 10. 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 used a stratified sampling design to draw a sample of "different" texts. In practice, at first texts were stratified according to their frequency of occurrence, hence, within each stratum, a simple random sample (without replacement) of them was selected. The sampling fraction was greater for the texts with higher occurrences because for these ones we desired a smaller error in estimates (Cochran, 1977). Table 7 shows various quantities used to calculate the approximate optimal sampling fraction within each single stratum.

Classes of occurrences	**Number of different texts**	**Hypothesised precision of autom. coding**	**Max error desired**	**Approximate optimal sample size**	**Sampling fraction**
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.

We do not have the class "1,001-10,000" because all its 41 different texts had a coding score equal to 10. This means that the system is able to code correctly "different" texts with the highest frequencies.

The sample of 938 texts was submitted to expert coders to evaluate if ACTR assigned them 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 obtained estimates can be found in the Table 8, with the corresponding quantity useful to calculate the 95%-confidence interval (last column of the table).

As it 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% $(= 75.77 + 5.19)$ with a probability approximately 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 here all texts (rather than a sample) were checked. For these classes 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 (%)**	**Values for conf. limits**
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 the 6,083 $(= 19{,}904 - 13{,}821)$ "different" texts coded with a score of 10 (all correctly coded) the overall estimated precision grows 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 one of the "different" texts, by considering the associated frequencies (Table 9).

Classes of occurrences	**"Different" texts**	**Original Texts**	**Estimated precision (%)**	**Values for conf. Limits**
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 a probability of 0.95. Here too, if we consider the 222,135 original text 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 perfectly 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 is possible to evaluate the precision of automated coding results with a high confidence. In should be noticed that here we did not introduce cost consideration about manual coders checking. This element should be taken into account in order to better calculate the optimal sample size when automatic coding is put in the flow of large surveys.

References

Appel M. and Hellerman E. (1983). Census Bureau experience with Automated Industry and Occupation Coding. In American Statistical Association, *Proceedings of Section on Survey Research Methods*, pages 32-40.

Chen B., Creecy R. and Appel M. (1993). Error control of automated industry and occupation coding. *Journal of Official Statistics*, vol. 9: 729-745.

Cochran W. G. (1977). *Sampling Techniques, 3rd ed.*. Wiley, New York.

Dumicic S. and Dumicic K. (1994). Optical reading and automatic coding in the Census '91 in Croatia. In *Conference of European Statisticians, Work Session on Statistical Data Editing,* Cork, Ireland 17-20 October, Working Paper n. 2.

Everitt B. S. (1977). *The Analysis of Contingency Tables.* Chapman and Hall, London.

Hellermann E. (1982). Overview of the Hellerman I&O Coding System. Internal document, US Bureau of the Census, Washington.

Lyberg L. and Dean P. (1992). Automated Coding of Survey Responses: an international review. In *Conference of European Statisticians, Work session on Statistical Data Editing*, Washington DC.

Kalpic D. (1994). Automated coding of census data. *Journal of Official Statistics*, vol. 10: 449-463.

Knaus R. (1987). Methods and problems in coding natural language survey data. *Journal of Official Statistics*, vol. 1: 45-67.

Massingham R. (1997). Data capture and Coding for the 2001 Great Britain Census. In *XIV Annual International Symposium on Methodology Issues*, 5-7 November, Hull, Canada.

Tourigny J.Y. and Moloney J. (1995). The 1991 Canadian Census of Population experience with automated coding. In United Nations Statistical Commission, *Statistical Data Editing*, 2.

Wenzowski M.J. (1988). ACTR – A Generalised Automated Coding System. *Survey Methodology*, vol. 14: 299-308.