Jussi Salmela
Viljo Kohonen

CHITAB - a "poor man's" shortcut to computer processing of linguistic data

1. Background. The primary purpose of the computer programme CHITAB is modest: we hope it to be of use mainly to individual linguists or small projects with limited resources, who want to cope with sizable corpuses involving delicate classifications. In such a case processing the data manually will soon become laborious. To be more adaptable to linguistic data processing, the present programme introduces some improvements over similar programmes already existing in various programme libraries (e.g., in HYLPS, in the Univac 1108 system of the University of Helsinki, and in the SPSS, for Dec 10). These improvements include the possibility for alpha-numeric coding, the use of up to ten "filter variables" to extract from the corpus precisely the desired variables for cross-tabulation, and the possibility to pick, out of the classifications of any variable, only the frequencies of any individual classificatory principles (e.g., codes 1,3,8,A,C, out of the total range from 1-F). These improvements mean a more economic use of the card space, and a more versatile use of the computer.

The basic idea in the system is that, instead of processing both the text and the coded symbols, only the symbols are fed into the computer. This solution naturally excludes certain kinds of research, such as vocabulary frequency studies, but it is adequate for frequency counts and cross-tabulations of, e.g., various semantic, syntactic and textual features. It is thus adaptable for a variety of research purposes. An important advantage of the system is that it is remarkably cheap: the punching of the cards is fairly quick and straightforward, one card can accommodate a large number of classifications, and the computer processing is also quick.

For the benefit of the individual researcher, who is frequently unsophisticated in computer technology, we have also attempted to make the actual use of the computer as simple as possible. Thus, in order to have the computer carry out the desired cross-tabulations, the user only needs

CHITAB - a "poor man's" shortcut to computer processing of linguistic data
Jussi Salmela, Viljo Kohonen
Proceedings of NODALIDA 1977, pages 82-86

to punch one or two cards: one card for the specification of the variables to be cross-tabulated (and giving the title of the table if desired), and another card to give the "filter" variables (if these are needed; cf. below).

2. Coding of the data. When analysing the primary material (text), the researcher transforms the features chosen for the study directly into a series of coding symbols. To do this he must have an idea of what he is looking for from the text, in the form of hypotheses to be tested or questions to which he wants to get quantitative answers. The codings are entered manually into primary matrices, according to a specific coding plan. The development of such a plan frequently involves pilot studies with experimental data. For computer processing, the data are punched onto cards.

In the present programme, one variable can be given a field of max. 5 columns, and a record can comprise a maximum of 5 cards (i.e., 400 one-column variables); with the alpha-numeric coding, one column can accommodate some 40 different symbols. In most cases, however, some 15-20 subclassifications will be enough, and one column will thus suffice. The first few columns will be two-or-more column variables, for an exact identification of the data (e.g., text/page/line, or consecutive numbering). We have not adopted the mnemonic coding symbols used, e.g., in MAMBA,[1] because the one-digit alpha-numeric symbols are more economical. In practice, we have noticed that the researcher can (and does) memorize quite a detailed coding plan with "decontextualized" symbols within the first few days (or weeks) of intensive coding work. This relieves him from constant checking of the symbols and thus speeds up the coding process. In the presentation of the results one naturally has to refer to the original text for relevant examples.

3. Checking and correcting of the data. In addition to the verifying punching, the programme provides four possibilities for the checking of the data. (1) The checking of the min. and max. limits of codings in each variable (e.g., from 1 to F; anything above F must be an error) reveals errors that are typically due to punching (these can be eliminated by verifying punching, though this is laborious in a large corpus). (2) The programme also gives the totals of coded and lacking symbols in each variable.

---

1 Ulf Teleman, Manual för grammatisk beskrivning av talad och skriven svenska, Lund 1974.

These are useful in cases in which the researcher knows that certain variables should have the same total frequencies (cf. check 3), and in extreme cases in which the researcher is reasonably sure that given variables should contain coding in all or very few records. (3) With interrelated variables, the programme can be used to check simultaneous presence/absence of coding in any combination of variables. For example, if the coding plan has four columns for different properties of the subjects of sentences (such as length, structure, givenness and position), all of these must have some coding or be blank. The computer prints out the records showing lack of agreement. (4) Further, "impossible" cross-tabulations can also be used to spot errors. Thus, for example, if passive sentences are entered in column X with codings 3,4,6, and their subjects are analysed in a subsequent column Y,1-8, a tabulation of the identification variables with X=3,4,6 and Y=blank/Ø (i.e., no coding) as the filter variables will give a list of records in which the subject properties had not been entered into the matrix. Errors in (3) and (4) are thus frequently due to the researcher forgetting to enter the relevant information in all the places; such errors cannot be revealed by verifying punching. With (4) one can even get inside the total frequencies of the variables, by specifying parts of them to be checked for agreement. Finally, the actual data cross-tabulations will, of course, serve as further checks in cases of impossible or meaningless contingencies. The programme can be used to print out the dubious records for checking and eventual correction. The erroneous records will be replaced on the magnetic tape by the corrected ones.

4. The cross-tabulation programme can be utilized for the following purposes: (1) extraction of data according to the desired specifications or their combinations ("filters"), (2) cross-tabulation of any two variables against each other, (3) calculation of the Chi square test and the contingency coefficient, and (4) calculation of the relative frequencies by the totals of the rows and columns, and the sum total.

(1) The maximum of the filter variables is 10. These are given on a separate card (or more than one card), after the card specifying the variables to be cross-tabulated. An example of a table request thus looks as follows:

```
16, 28 = 1  First Table
    5 = 1-4,5,9,A   6 = 3,4   25 = 1-8,F
```

This means that variables 16 and 28 will be cross-tabulated against each other, and the data is extracted from that part of the corpus which is specified by the above values of variables 5, 6 and 25. If only portions of

the variables are desired for the tabulation, these must be given in the filter variables (e.g., giving 16 = 1,3,5,A on the second card in the above example). To save computer time, the programme has a "peeping device", i.e., if precisely the same filters are used in the following table, an auxiliary file is formed of the relevant records. This file is kept until a new combination of the filter variables is read. To make use of this possibility to save time, the researcher should group his table requests in such a way that the tables to be run from precisely the same sub-part of the corpus are in a consecutive order. This is not necessary, however, if it is more desirable to group the runs in some other way. A consequence of this limitation is that the tables to be run without any filter variables (i.e., from the whole corpus) must be run first.

(2) For the cross-tabulations, the printout format provides 25 columns and 155 rows, if needed. These limitations should be taken into account when grouping the row and column specifications. The variable given first in the table request is always interpreted as the row variable, and the second as the column variable. The frequency table is automatically supplemented by the percentages of the row and column totals out of the sum total, for quick reviews. By way of speeding up the programme, the testing of the filter variables and the up-dating of the frequency tables are both done as a uniform binary search. The programming language is FORTRAN 4.

(3) The calculation of the Chi square test is optional. It is done according to the formulas given in Siegel (1956).[1] Before applying the formulas, the programme checks that the conditions for the calculation of the Chi are satisfied by the contingency table. In the negative case, the programme prints out the reason for the inapplicability of the test. As this module also accepts contingency table data from the cards, it can be used as an independent unit for calculating the test, after possible re-groupings of the tabulated frequency data. When the Chi test is applicable, the programme also calculates the value of the contingency coefficient (C), as a measure of the degree of association between the two variables. These are given immediately below the frequency tables, with the df value.

(4) The calculation of the percentages by the totals of the rows, columns and the sum total is also optional, and it is possible to choose any of these.

---

1 Sidney Siegel, Nonparametric Statistics, New York 1956.

5. Concluding remarks. The members of the Text Linguistics Research Group are working on syntactic data collected from Finnish, Swedish and English, to be processed on the CHITAB. Kohonen has collected a corpus of some 4,000 clauses from Old and Early Middle English (between ca. 1000 and 1200) for a study of the development of OE word order. The results will be published in the research reports of the group in 1978. A documentation of the programme and the coding plans developed for Finnish and Swedish, with some preliminary findings, will appear in Erik Andersson (ed.), Working Papers on Computer Processing of Syntactic Data, Abo 1978.

Our future plans for the development of the programme include a module that can be used for matching a dependent clause to its matrix clause and forming a separate file of the matrix clauses, for analyses of desired features in them. This module could also be used in contrastive studies, when comparing translations with the originals. Another improvement could be added to the tables, where the decontextualized symbols could be automatically replaced, if desired, by mnemonic 4-character titles, to be fed into the computer separately. This would still preserve the ease and economy of the input, while making the interpretation of the tables more convenient. A further improvement could also be added to the "peeping device": instead of taking the full records into the auxiliary file (when precisely the same part of the data is going to be used for several consecutive tables), only the relevant variables, i.e., those actually needed for the tables requested, would be extracted from the data. This would again speed up the programme.

The programme is now available in the Dec 10 system of the University of Turku. We are planning to get it into the Univac 1108 system of the University of Helsinki, with terminals all over Finland.[1]

1 The designing of the operations carried out by the programme has been done jointly by the two authors, while all the technical programming work has been done by Jussi Salmela. If somebody is interested in getting a copy of the programme he should contact Jussi Salmela.