Grotjahn, Rüdiger. (Hrsg.). (1996). Der C-Test. Theoretische Grundlagen und praktische Anwendungen (Bd. 3, S. 401–418). Bochum: Brockmeyer

Gerhard Koller and Rosemary Zahn

Computer based construction and evaluation of C-tests

A problem area that confronts most organisers of foreign language courses, particularly at university level, is that of placing students in the appropriate course. Placement tests for large numbers usually require considerable resources both as regards personnel and facilities. The Language Centre of the University of Erlangen-Nürnberg has developed a programme for a computerised C-test by means of which the placement procedure is almost fully automated.

With this programme, the test designer's task consists of no more than choosing and typing the texts to be mutilated. The tests are scored electronically and the results evaluated statistically. The programme can be used at an individual personal computer or in a network. The programme is user-friendly and flexible. Its potential exceeds the limits of C-test batteries alone and can be used for a wide variety of fill-in tests at all levels.

1. Introductory Remarks

Whether and to what extent a person is capable of decoding acoustic and visual language signals correctly when there is interference in the communication channel in question (through noise, illegible writing etc) depends to a great extent on the person's competence in the language concerned. Since interference of this nature is a common phenomenon, communication can only function effectively because language systems have high levels of redundancy. Studies have shown that as much as 75% of written language is redundant.

Linguistic competence therefore presupposes the ability to decode language signals correctly even when there is interference in the communication channels. Since native-speakers of a language usually have no difficulty in decoding imperfect language input, this ability can be regarded as one way of measuring competence in a given language (cf. Klein-Braley, 1985). One method of measuring this skill consists of reconstructing texts in which elements have been deliberately omitted or mutilated. The programme under discussion here enables the user to construct language tests based on this principle of *reduced redundancy*.

The C-test is a special type of cloze test developed by Klein-Braley and Raatz. Its canonical format is as follows (cf. Raatz 1985a, pp. 15f.):

- It consists of approx. 4 text segments.
- The first sentence of a text remains unchanged in order to provide a minimum of contextual information.
- Beginning with the second word of the second sentence, the second half of this and every successive second word is deleted (mutilated). 25 word forms are mutilated per text segment.
- In words with an uneven number of letters, one more letter is deleted. Word forms consisting of only one letter (e.g. Engl. *I* and *a*) are ignored. Numbers and figures are marked for non-deletion.
- Once the required number of deletions has been reached, the remainder of the text is left unchanged.

One of the major advantages of the C-test is that it apparently guarantees high levels of validity and reliability while being relatively simple to design and evaluate. In a series of examples Grotjahn (1987) has shown that in many respects the C-Test principle is not as apparently simple as it may seem and requires further analysis in several instances.

If a relatively large group of test-takers need to be tested on their language proficiency level at short notice in order, for example, to place them in university language courses, the personnel factor involved still remains a major consideration despite the administrative advantages specific to the C-test.

In order to enrol in language courses for non-language majors at the Language Centre of the University of Erlangen-Nürnberg, for example, as many as 300 students take part in a placement test for English alone shortly before the start of the new semester. Since the evaluation of these tests takes place under great time pressure, problems have arisen in the past which, in view of staff shortages, have been hard to resolve. This is the background situation which has led to the development of a computer programme intended to reduce manual correction work to a minimum and which constitutes the content of this report[1].

[1] As far as we know there are no comparable programmes for constructing C-tests currently on the market. Germann (1996) has developed an automated procedure for constructing C-tests based on *Word for Windows*.

2. The computer program

2.1 Preparation of the test texts

The test constructor intent on designing nothing more than a standard test battery has only to select the necessary texts for a test and to collate them in such a way that they can either be scanned into or read by a computer.

Any text processing programme containing an option which will save a typed text as an ASCII file is capable of this. All the current programmes on the market such as WORD, Word-Perfect etc. are appropriate systems with ASCII file options.

The text can be typed in irrespective of layout considerations because the layout will be dealt with automatically by the programme.

2.2 C-test variants

The programme is suitable for a variety of target groups. On the one hand the programme has been designed for a user who is content to apply only the standard prescriptions of the C-test. And in this case, all the processing steps which are part of the production of a C-test will be carried out by the computer. The test designer's task will essentially be limited to selecting the texts to be used. On the other hand, rather more adventurous or demanding users will find that it offers a great variety of presentation options.

The following alternative processing variants are available:

(a) C-test: Standard

The text is processed according to the standard C-test requirements as described above in 3.

(b) C-Test: Randomisation

There has been much controversial discussion as to whether C-tests should indicate the length of the word for test candidates by indicating the number of letters deleted or not (cf. Meißner-Stiffel & Raatz, 1996). In addition to the standard programming on a 1:1 basis in which the length of the word can be deduced from the number of dashes provided, the programme offers two alternatives. In the randomised variant the text is again prepared according to the C-test standard basic test format. The number of dashes indicating a mutilated word is randomised but is at least as high as the number of deleted letters.

(c) C-Test: one gap

To increase the level of difficulty, only one continuous dash is inserted here in each of the mutilated words. The test candidate thus only receives information as to where a deletion has occurred. Otherwise the procedure is as above.

2.3 Manual modification options

The programme offers the following options if the test designer wants to modify the standard test format:

(a) The modified starting point

It is not always advisable or desirable to start by deleting the second word of the second sentence, as Grotjahn (1987, p. 223) has demonstrated. In a case such as this the programme offers two different ways of varying the procedure:

- The passages at the beginning of a text which are not to be deleted can be marked for non-deletion.
- The text passage which is to be mutilated can be marked as a block. It is also possible to mark or define several passages in the text if the deletions are to be only partial or are not to be continuous throughout the text.

(b) The number of deletions

Views on the number of deletions that should occur in each text segment vary greatly in publications on C-tests to date. Raatz (1985) suggests that 20 items should be deleted, whereas Grotjahn (1987) argues that this number might be increased somewhat in some special cases. Further investigation is required before it can be stated categorically whether it is sensible to lay down an exact number of deletions or whether the number will depend on specific features determined by individual texts or indeed languages. The programme takes this problem into account by permitting manual definition of the number of items per test segment.

(c) The range of deletions

The degree of variance and lack of consensus as to how much of a word and what part of the word should be deleted is almost as great as the controversy about the number of deletions that should be made. Following the suggestions put forward by Köberl and Sigott (1994), in addition to the standard deletion procedure (the second half of a word), the following variants are proposed:

- Two thirds of a word are deleted;
- A word is deleted except for the initial letter.

(d) Alternative solutions

The question whether alternative solutions which do not change the meaning of a text significantly should be permitted or whether only the solution contained in the original text should be regarded as acceptable has also given rise to differences of opinion (cf. Grotjahn, 1987, pp. 222f.). The programme is flexible on this point too and allows for alternative solutions where necessary.

A decision on which alternatives are acceptable can only sensibly be taken after the test has been administered. The programme consequently lists all the variants given on each item separately, thus providing the necessary data on which the tester can base his decisions as regards acceptability. Differences in acceptable orthographic variants can be included here, for example German *Photographie/Fotografie*, or AE/BE differences such as *gray/grey*. In a second scoring re-run the programme then accepts these variants as correct.

(e) The treatment of proper names

Word forms which are not to be deleted such as proper names can be marked for non-deletion manually.

(f) The number of test segments

The number of test segments the programme can deal with is virtually unlimited but a text cannot be longer than one screen page. If a longer text needs to be scored, it must be broken down into screen lengths.

(g) The level of difficulty

A range of different parameters can be applied which will influence a test's level of difficulty:

- Reference has already been made to the different methods of presenting the gaps, which in turn will influence the facility indices.
- In addition to these the tester can decide upon and set a time limit. The test time is shown digitally on the screen while the test-taker does the test and once this time limit has expired the programme will be automatically interrupted.
- It is also possible to make the test more difficult by preventing the test-taker from moving the cursor up and down the screen in a test segment, which in effect means he cannot revert to passages he has completed or passed over to alter them. (The cursor then only progresses forward).

(h) Apostrophe

In the original form of the C-test word combinations with an apostrophe are treated as constituents of a word form. But the logic of treating *l'homme* as one word whereas *la femme* is seen as two is a little puzzling to say the least. Reducing *dell'opera* to *dell'.....* , thus deleting a complete lexeme, equally seems to clash with the C-test principle (examples taken from Grotjahn, 1987). The programme is able to interpret the apostrophe as an element which conjoins as well as separates words.

(i) Weighting

In the basic configuration, each correct solution is allotted 1 point, but different test types and different items within them can be weighted differently. Weighting test types or items differently would be advantageous, for example, if a test consists of different test types that differ in length or level of difficulty. The items in a C-test could be differentiated according to structure words and meaning words. In order to do this, the tester would simply have to supply a list of the meaning or structure words.

2.4 Output

The programme delivers a series of score results printed in the form of an ASCII file. This data can subsequently be processed by any normal text processing system. For those who might be interested in processing the evaluation data themselves, an interface has been inserted. The following sample of data represents the current range of test information the programme offers. The names and scores are fictitious and are only intended to serve as an example of test output.

Table 1
Summary statistics for test segment 1

Text: Scientists	Test Segment 1
Total number of points:	25.00
Range of points:	4.00 – 25.00
Mean:	19.00 pts (69.63%)
Standard deviation:	4.74 pts (18.94%)
Variance:	22.43
Difficulty index:	0.76
Discrimination index:	0.34

Table 2
List of test takers: alphabetical (test segment 1)

Rank	Points		Name, First Name
	abs.	perc.	
40	11	25.00	Aldebert, Dagmar
29	14	56.00	Andrjuscenko, V.M.
11	22	88.00	Arnolt, Gaby
15	21	84.00	Bakonyi, H.
20	20	80.00	Black, J.W.
⋮	⋮	⋮	⋮
11	22	88.00	Rieger, Florian
6	23	92.00	Scharrer, Gerhard
2	24	96.00	Schütz, Helmut
34	18	40.91	Streber, Anne
32	12	48.00	Zeisser, Christian

Table 3: List of test-takers: rank order

Rank	Points		Name, First Name
	abs.	perc.	
1	25	100.00	Lappus, Erna
2	24	96.00	Danilenko, V. P.
2	24	96.00	Domus, Christa
2	24	96.00	Schütz, Helmut
2	24	96.00	Huber, Fritz
6	23	92.00	Eldrige, R. C.
⋮	⋮	⋮	⋮
37	17	38.64	Eaton, H. S.
39	14	31.82	Fries, Ch. C.
40	11	25.00	Aldebert, Dagmar
41	5	20.00	Bongers, H.
42	4	16.00	Hartmann, Irene

Table 4
Score List: Whole test (alphabetical)

Rank	Name	Score (total)			Score: TS 1		Score: TS 2		Score: TS 3		Score: TS 4	
		abs.	perc.	SD(%)	perc.	abs. (59*)	perc.	abs. (25)	perc.	abs. (25)	perc.	abs. (28**)
41	Alldebert, Dagmar	57	39.45	(26.02)	82.50	(33)	25.00	(11)	36.00	(9)	14.29	(4)
30	Andrjuscenko, V.M.	110	74.58	(17.08)	96.61	(57)	56.00	(14)	60.00	(15)	85.71	(24)
19	Arnolt, Gaby	123	86.86	(10.17)	100.00	(59)	88.00	(22)	88.00	(22)	71.43	(20)
16	Bakonyi, H.	125	88.43	(6.72)	100.00	(59)	84.00	(21)	84.00	(21)	85.71	(24)
6	Black, H.W.	130	93.00	(8.19)	100.00	(59)	80.00	(20)	92.00	(23)	100.00	(28)
...	...	...	...	...	...	...	...	...	...	...	...	...
20	Rieger, Florian	85	86.05	(3.88)	90.48	(19)	88.00	(22)	80.00	(20)	85.71	(24)
2	Scharrer, Gerhard	132	95.00	(5.20)	100.00	(59)	92.00	(23)	88.00	(22)	100.00	(28)
7	Schütz, Helmut	128	92.58	(4.34)	96.61	(57)	96.00	(24)	92.00	(23)	85.71	(24)
33	Streber, Anne	95	71.21	(19.03)	92.50	(37)	40.91	(18)	80.00	(20)	71.43	(20)
32	Zeisser, Christian	90	73.97	(16.77)	92.50	(37)	48.00	(12)	84.00	(21)	71.43	(20)

* Total results from optional item weighting factor – included here as example
** Total results from optional test type weighting factor – included here as example

Table 5
Score List: Whole test (rank score)

Rank	Name	Score (total)			Score: TS 1		Score: TS 2		Score: TS 3		Score: TS 4	
		abs.	perc.	SD(%)	perc.	abs. (59)	perc.	abs. (25)	perc.	abs. (25)	perc.	abs. (28)
1	Lappus, Erna	136	99.00	(1.73)	100.00	(59)	100.00	(25)	96.00	(24)	100.00	(28)
2	Eldrige, R. C.	132	95.00	(5.20)	100.00	(59)	92.00	(23)	88.00	(22)	100.00	(28)
2	Scharrer, Gerhard	132	95.00	(5.20)	100.00	(59)	92.00	(23)	88.00	(22)	100.00	(28)
4	Pfeiffer, Gustav	112	94.38	(4.68)	97.50	(39)	88.00	(22)	92.00	(23)	100.00	(28)
5	Domis, Christa	130	93.43	(5.28)	100.00	(59)	96.00	(24)	92.00	(23)	85.71	(24)
:	:	:	:	:	:	:	:	:	:	:	:	:
38	Fuminka, R. M.	62	63.76	(13.74)	76.19	(16)	60.00	(15)	76.00	(19)	42.86	(12)
39	Harmann, Irene	76	60.68	(28.44)	85.00	(34)	16.00	(4)	56.00	(14)	85.71	(22)
40	Meier, Johannes	70	56.13	(34.89)	92.50	(37)	56.00	(14)	76.00	(19)	0.00	(0)
41	Aldebert, Dagmar	57	39.45	(26.02)	82.50	(33)	25.00	(11)	36.00	(9)	14.29	(4)
42	Bongers, H.	50	38.21	(20.59)	70.00	(28)	20.00	(5)	20.00	(5)	42.86	(12)

Table 6
Mean, standard deviation and variance for the whole test and test segments

Test seg.	max.	Range	M	SD	DI	Dis I	Ranks
TS 1	59	13 – 59	44.76	13.83	0.76	0.17	42
TS 2	25	4 – 25	19.00	4.74	0.76	0.34	42
TS 3	25	5 – 24	19.55	4.11	0.78	0.31	42
TS 4	28	0 – 28	22.67	6.21	0.57	0.30	42
Whole test	137	22 – 136	105.98	22.23	0.67	0.29	42
Reliability (Cronbach's Alpha): 0.61							

Table 7
Correlations (Whole test with individual test segments and intercorrelations with test segments)

TS 1 – 2 :	0.47
TS 1 – 3 :	0.51
TS 1 – 4 :	0.24
TS 2 – 3 :	0.53
TS 2 – 4 :	0.15
TS 3 – 4 :	0.22
WT – TS 1 :	0.67
WT – TS 2 :	0.81
WT – TS 3 :	0.76
WT – TS 4 :	0.55

TS = Test Segment
WT = Whole Test

Table 8
Distribution over all test segments: Scoring of the difficulty index and discrimination per item

(a) Discrimination Index

Range		Number	
from	to	abs.	perc.
-1.00	-0.90	0	0.00
-0.90	-0.80	0	0.00
-0.80	-0.70	0	0.00
-0.70	-0.60	0	0.00
-0.60	-0.50	0	0.00
-0.50	-0.40	0	0.00
-0.40	-0.30	0	0.00
-0.30	-0.20	0	0.00
-0.20	-0.10	0	0.00
-0.10	0.00	0	0.00
0.00	0.10	27	31.76
0.10	0.20	6	7.06
0.20	0.30	10	11.76
0.30	0.40	19	22.35
0.40	0.50	6	7.06
0.50	0.60	9	10.59
0.60	0.70	3	3.53
0.70	0.80	2	2.35
0.80	0.90	2	2.35
0.90	1.00	0	0.00
1.00		1	1.18
Total number of items: 85			

(b) Difficulty Index

Range		Number	
from	to	abs.	perc.
0.00	0.10	8	9.41
0.10	0.20	0	0.00
0.20	0.30	2	2.35
0.30	0.40	0	0.00
0.40	0.50	2	2.35
0.50	0.60	1	1.18
0.60	0.70	10	11.76
0.70	0.80	8	9.41
0.80	0.90	15	17.65
0.90	1.00	23	27.06
1.00		16	18.82
Total number of items: 85			

Table 9
List of the variants offered (with frequencies)

	interview		reached		sleepily
31	inte'rview	30	rea'ched	41	slee'ping
14	inte'	12	rea'	16	slee'pily
6	inte'grate	5	rea'lity	4	slee'
2	inte'llige	5	rea'ding	2	slee'pely
1	inte'rtain	4	rea'lise	2	slee'ply
1	inte'rest	2	rea'lize	1	slee'pish
1	inte'rests	1	rea'cion	1	slee'pped
1	inte'druce	1	rea'sult	1	slee'zily
1	inte'nsify	1	rea'c		
1	inte'ntion	1	rea'dout		
1	inte'rwio	1	rea'lies		
1	inte'rwiew	1	rea'sons		
1	inte'r	1	rea'veld		
1	inte'viewe	1	rea'lisd		
1	inte'nding	1	rea'cted		
1	inte' rfere	1	rea'lled		
1	inte'llect				
1	inte'rvise				
1	inte'rres				

Table 10
Number of correct solutions per item (total of all test-takers; according to rank thirds: top – middle – bottom; difficulty index; discrimination index

	Item	Total		Top 1/3		Middle 1/3		Bottom 1/3		Difficulty	Discrim.
		abs	perc	abs.	perc.	abs.	perc.	abs.	perc.	index	index
1	If	30	17.43	13	92.86	9	64.29	8	57.14	0.71	0.36
2	continues	25	59.52	13	92.86	6	42.86	6	42.86	0.60	0.50
3	the	40	95.24	14	100.00	14	100.00	12	85.71	0.95	0.14
4	rate	25	59.52	11	78.57	9	64.29	5	35.71	0.60	0.43
5	of	26	61.90	13	92.86	7	50.00	6	42.86	0.62	0.50
6	in	42	100.00	14	100.00	14	100.00	14	100.00	1.00	0.00
7	atmosphere	39	92.86	14	100.00	13	92.86	12	85.71	0.93	0.14
8	cause	39	92.86	14	100.00	14	100.00	11	78.57	0.93	0.21
9	onset	28	66.67	13	92.86	12	85.71	3	21.43	0.67	0.71
10	an	41	97.62	14	100.00	14	100.00	13	92.86	0.98	0.07
11	age	29	69.05	14	100.00	12	85.71	3	21.43	0.69	0.79
12	about	38	90.48	14	100.00	13	92.86	11	78.57	0.90	0.21
13	years	41	97.42	14	100.00	14	100.00	13	92.86	0.98	0.07
14	This	33	78.57	14	100.00	8	57.14	11	78.57	0.79	0.21
15	the	39	92.86	14	100.00	14	100.00	11	78.57	0.93	0.21
16	reached	10	23.81	7	50.00	3	21.43	0	0.00	0.24	0.50
17	leading	29	69.05	13	92.86	11	78.57	5	35.71	0.69	0.57
18	at	36	85.71	13	92.86	13	92.86	10	71.43	0.86	0.21
19	US	19	45.24	11	78.57	3	21.43	5	35.71	0.45	0.43
20	flight	24	57.14	10	71.43	9	64.29	5	35.71	0.57	0.36
21	at	35	83.33	13	92.86	13	92.86	9	64.29	0.83	0.29
22	recent	27	64.29	13	92.86	9	64.29	5	35.71	0.64	0.57
23	attended	29	69.05	13	92.86	9	64.29	7	50.00	0.69	0.43
24	researchers	35	83.33	14	100.00	13	92.86	8	57.14	0.83	0.43
25	scientists	39	92.86	14	100.00	14	100.00	11	78.57	0.93	0.21

2.5 Administering the test

Our aim was not only to construct the test with the help of the computer but also to administer it with a minimum of personnel and to have it evaluated by the computer as well.

The candidates for the test are fed through the networked computer pool in groups of 25 – 30.[2] After a brief introductory and practice phase (for computer non-initiates), the tests are done on the computer screen. (This requires minimal computer skills. Apart from the keyboard letters, only the cursors, the delete and return keys are used). The results are then evaluated electronically.

Since networked computer pools are not available everywhere, the programme can also be used at individual PCs. The data can be loaded onto individual PC hard disks and at the end of the test collated and evaluated.

The computer programme provides the tester not only with the data he or she needs to set up the courses the candidates for the test have applied for but also a mass of statistical evaluations.

The programme has the necessary flexibility to be able to change a series of test parameters with great ease, which in turn means that the special characteristics of the C-test principle can be examined under a variety of conditions even with large groups of test candidates. This is an aspect which could play a significant role in developing this type of test further.

If only one computer is available for the construction of a test, the programme we are presenting here also offers the option of designing a "paper version" of a C-test. The test is compiled by the computer and the finished version of the test can be copied into a file by means of the *export option* – or with the help of any word processing programme – where it can be further processed as a paper or hard copy.

A whole range of articles (cf. Klein-Braley, 1985, p. 18 for example and Raatz, 1985) have already commented on the various uses of the C-test. The programme under discussion here is not only capable of compiling and carrying out tests, but rather it also contains a self-learning option by means of which the learner receives an analysis of his performance when he has completed the test on the PC, and which shows on the screen how he has

[2] The group size is determined by the size of the computer pool, in this case 30.

performed in the test. Thus the programme could also be used for follow-up work or further analysis of texts used in class as well as in self-learning processes independent of a teacher.

3. Summary and outlook

The programme was written in Turbo Pascal and implemented under the DOS system[3]. Its main features are summarized in Figure 1.

Currently a WINDOWS version of the programme is in progress. This will have the advantage that, in addition to offering better graphic design facilities, it will be easier to work on tests which are based on alphabets other than the Latin one. This also applies to languages which are not written from left to right. In addition, we have plans to modify the programme to include further test types. Listening comprehension tests could be constructed by combining sound and video. In combination with a dictation programme which is currently being developed, we aim to create an authoring system which will enable the language teacher who has no programming experience to develop supplementary course materials or computer assisted individualised language learning materials with a minimum of effort.

[3] The programming was carried out by Jan Stiller, whom we would like to thank most warmly for all his hard work and cooperation.

Figure 1
Main features of the computer program

The computerised construction and evaluation of C-tests:

The programme's performance/features
- Automatic construction of a C-test with standard parameters;
- Numerous additional manual variants:
  - Variable starting-point;
  - Variable number and length of deletions;
  - Accepts alternative solutions (after the test and before the evaluation!);
  - Several alternative ways of designing gaps;
  - Text passage to contain deletions can be marked;
  - Proper names etc. can be manually marked for non-deletion (numbers automatically marked for non-deletion);
- Can be used in network or at single PC.

Evaluation
- Automatic correction;
- List of test-takers:
  - alphabetical;
  - in rank order (acc. test segment and/or whole test);
- Test population rank order in individual test segments;
- Mean, standard deviation and variance for whole test and test segments;
- Difficulty rating per test segments;
- Correlation:
  - whole test with individual test segments;
  - intercorrelations of test segments;
- Lists of variants on items;
- Number of correct solutions per item:
  - total test population;
  - ranked in thirds – top – middle – bottom;
- Calculation of difficulty rating and discrimination per item;
- Time factor per item.

...

For further information please contact:
Sprachenzentrum der Universität Erlangen-Nürnberg
c/o Prof. Dr. Gerhard Koller, Bismarckstraße 1, D-91054 Erlangen
E-Mail: gdkoller@phil.uni-erlangen.de – Fax: 09131/852194

References

Germann, Ulrich. (1996). C-Tests automatisch erstellen – mit Word für Windows 6.0. In Rüdiger Grotjahn (Ed.), *Der C-Test: Theoretische Grundlagen und praktische Anwendungen* (Vol. 3). Bochum: Brockmeyer.

Grotjahn, Rüdiger. (1987). How to construct and evaluate a C-test: A discussion of some problems and some statistical analyses. In Rüdiger Grotjahn, Christine Klein-Braley & Douglas K. Stevenson (Eds.), *Taking their measure: The validity and validation of language tests* (pp. 219-253). Bochum: Brockmeyer.

Grotjahn, Rüdiger. (Ed.). (1992). *Der C-Test: Theoretische Grundlagen und praktische Anwendungen.* (Vol. 1). Bochum: Brockmeyer.

Grotjahn, Rüdiger. (Ed.). (1994). *Der C-Test: Theoretische Grundlagen und praktische Anwendungen.* (Vol. 2). Bochum: Brockmeyer.

Klein-Braley, Christine. (1985). Reduced redundancy as an appoach to language testing. In Klein-Braley & Raatz (1985), 1-13.

Klein-Braley, Christine & Raatz, Ulrich. (Eds.). *Fremdsprachen und Hochschule 13/14: Thematischer Teil: C-Tests in der Praxis.* Bochum: AKS.

Köberl, Johann & Sigott, Günther. (1994). Adjusting C-test difficulty in German. In Grotjahn (1994), 179-192.

Meißner-Stiffel, Marion & Raatz, Ulrich. (1996). _____ oder _ _ _ _? Zwei Grundlagenuntersuchungen zum C-Prinzip bei L1-Lernern. In Rüdiger Grotjahn (1996) (Ed.), *Der C-Test. Theoretische Grundlagen und praktische Anwendungen* (Bd. 3). Bochum: Brockmeyer.

Raatz, Ulrich. (1985a). Tests of reduced redundancy – the C-test, a practical example. In Klein-Braley & Raatz (1985), 14-19.

Raatz, Ulrich. (1985b). C-Tests im muttersprachlichen Unterricht. In Klein-Braley & Raatz (1985), 66-71.

Raatz, Ulrich & Klein-Braley, Christine. (1985). How to develop a C-test. In Klein-Braley & Raatz (1985), 20-22.

Grotjahn, Rüdiger. (Hrsg.). (1996). Der C-Test. Theoretische Grundlagen und praktische Anwendungen (Bd. 3, S. 419–434). Bochum: Brockmeyer

Ulrich Germann

C-Tests automatisch erstellen – mit Word für Windows 6.0

Der folgende Beitrag stellt ein Programm zur automatischen Erstellung von C-Tests unter Word für Windows 6.0 vor. Dazu gibt er zunächst einen Überblick über Modifikationen des 'klassischen' Tilgungsverfahrens, die von verschiedenen Autoren im Rahmen der C-Test-Forschung der letzten Jahre vorgeschlagen wurden. Anschließend wird vor diesem Hintergrund das Programm in seinen wichtigsten Leistungsmerkmalen beschrieben.

1. Einleitung

Seit seiner Einführung durch Klein-Braley und Raatz im Jahre 1981 hat sich der C-Test in zahlreichen Untersuchungen als reliables und valides Instrument zur Messung der globalen Sprachleistung von Fremdsprachenlernern und Muttersprachlern erwiesen. Darüber hinaus zeichnet er sich durch ein angesichts seiner Zuverlässigkeit und Aussagekraft sehr ökonomisches Erstellungsverfahren aus. In dem von Raatz & Klein-Braley (1985) detailliert beschriebenen 'klassischen' Verfahren wird zunächst von sechs Test-Texten ausgegangen. In jedem Text wird, beginnend mit dem zweiten Wort des zweiten Satzes, die hintere Hälfte jedes zweiten Wortes getilgt. Die Länge des Wortes wird dabei in Buchstaben gemessen. Ist die Anzahl der Buchstaben eines Wortes ungerade, wird ein Buchstabe mehr getilgt als stehenbleibt. Aus *Hund* (4 Buchstaben) wird also *Hu*____, aus *Katze* (5 Buchstaben) *Ka*____. Wörter mit nur einem Buchstaben bleiben dabei unberücksichtigt. Pro Text werden auf diese Weise zwanzig Items erstellt. Am Ende jedes Texts bleibt ein Stück Text unversehrt erhalten, um ausreichenden Kontext zu gewährleisten. Schließlich scheiden diejenigen beiden Texte aus, die am wenigsten den Anforderungen an einen Test-Text genügen, d.h. die zu leicht oder zu schwer sind oder am wenigsten mit dem Gesamtergebnis korrelieren. Der fertige C-Test besteht also aus vier einzelnen Texten mit jeweils zwanzig Items.

Obwohl dieses Verfahren an sich sehr einfach ist, ist die manuelle Erstellung eines C-Tests trotzdem mühselig und fehlerträchtig. Zur Erstellung