

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/225371548>

SATS: Sequential Analysis of Transcripts System

Article in Behavior Research Methods · May 1990

DOI: 10.3758/BF03209829

CITATIONS

13

READS

46

2 authors:



Paul Yoder

Vanderbilt University

185 PUBLICATIONS 5,548 CITATIONS

[SEE PROFILE](#)



Jon Tapp

Vanderbilt University

19 PUBLICATIONS 550 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Language skills and pragmatic function of communication acts in young children with autism spectrum disorder [View project](#)



Vocal contingency and qualities of speech in young children with autism spectrum disorder [View project](#)

SATS: Sequential Analysis of Transcripts System

PAUL J. YODER and J. T. TAPP, SR.
Vanderbilt University, Nashville, Tennessee

Sequential analysis and transcripts are natural partners. Both are frequently used by researchers interested in aspects of social interaction (e.g., child language, parent-child interaction, husband-wife interaction), and the utterances or turns in the transcript are frequently the events that are subjected to sequential analysis. However, transcription and sequential analysis are time-consuming, labor-intensive tasks. We have therefore created the Sequential Analysis of Transcripts System (SATS) to allow flexible and efficient sequential analysis of coded transcripts.

SATS requires that the input data be a single stream that uses repeatable behavior codes. The form of the input data may be a coded transcript, which will be translated by the SATS program STRAN, or the data format of a STRAN output file, a vertical stream of digit codes representing events.

SATS provides the relative newcomer to sequential analysis with an easy-to-use set of programs that automatically provide information useful for analyzing many research questions involving a sequential level of analysis. Such research questions include the following: (1) "Is the frequency with which one actor's behavior follows another actor's behavior greater than expected by the two behaviors' base rates?" (2) "Is the sequential dependency of one target behavior/given behavior pattern greater than that of another pattern?" (3) "Is the sequential dependency of a specified sequential pattern the same in two or more groups?" Readers who desire an introduction to sequential analysis are referred to Bakeman and Gottman (1986).

Hardware Requirements for SATS

The programs described in this paper are written in VAX FORTRAN and run under the VMS operating system on the DEC VAX 8800. Since Microsoft FORTRAN Compiler 4.0 or higher supports the VAX FORTRAN extensions, the programs can also be run on the IBM PCs and compatibles with smaller array sizes. The programs should work properly with an ANSI standard FORTRAN 77 compiler. Due to the extremely time- and memory-consuming nature of the string translating program (STRAN) and of lag sequential analysis (LAGSEQ), it is faster to run the programs on a VAX whenever possible. We have ported the LAGSEQ program to an IBM mainframe running CMS and this source code is also available.

This research was supported by NICHD Grants HD22812 and HD15052. Correspondence may be addressed to J. T. Tapp, Sr., Box 40 GPC, Kennedy Center, Vanderbilt University, Nashville, TN 37203.

Specifications of SATS

SATS consists of four computer programs and a system configuration file that contains several parameters used in the programs. The programs are described in detail in the next sections. The specifications of each of the four programs are provided in Table 1, and an overview of the SATS process is provided in Figure 1.

The Programs in SATS

STRAN. STRAN creates a vertical stream of integer codes from the coded aspects of the transcript file while leaving the transcript file intact. The integer codes are 1-5 digits long. To run STRAN, the program requires an ASCII transcript file with a speaker code in the first column, and a code translation file (see Appendix A).

Additional optional information about each utterance in the input transcript may be coded by the experimenter within square brackets (i.e., []). These minimum requirements are compatible with a widely used transcription format and summary level linguistic analysis program (Systematic Analysis of Language Transcripts; Miller & Chapman, 1983).

The code translation file indicates which experimenter-coded aspects of the utterances are to be analyzed and which integer codes represent these aspects. This translation file must provide an integer code for every event in the transcript. For each event type, the following are identified: (1) the speaker or other event identification code (required); (2) the codes with square brackets that indicate the coder's judgment about some aspect of the event (optional); (3) the integer code that represents the bracketed and/or speaker code (required); and (4) the label describing the event (optional).

The code translation file allows the experimenter to select which experimenter-coded aspects of the utterances

Table 1
Software Specifications for the Programs in SATS

Program	Function	Array Limitations
STRAN.FOR	Translates transcript to integer codes	5,000 transcript lines of up to two 80-character strings per line 50 80-column translation lines 20 SALT codes per translation line
KAPPA.FOR	Computes interobserver agreement estimates	30,000 input events 50 unique codes
LAGSEQ.FOR	Computes lag sequential analysis statistics	30,000 input events 50 unique codes 10 lags
POOL.FOR	Pools translated files	100 files

Note—The limitations depend on the array sizes. Limitations presented are as the program now stands; they can be modified according to the needs of a particular project.

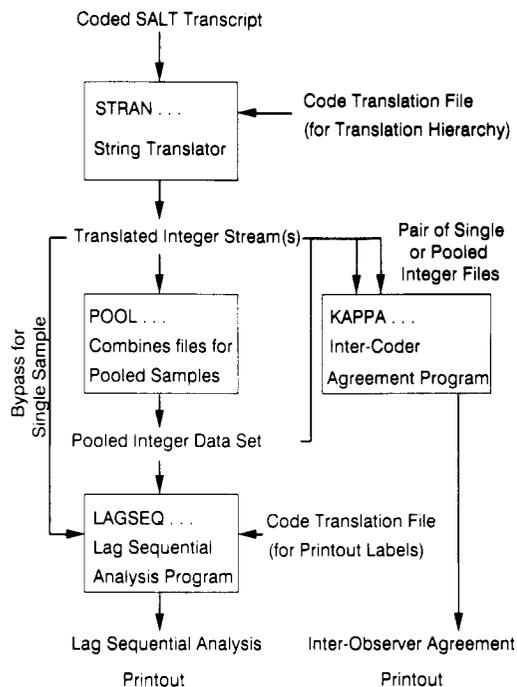


Figure 1. Overview diagram of the SATS process.

are to be analyzed sequentially. For example, we code two aspects of adult utterances: whether adult utterances continue a conversational topic or not, and whether they are questions or nonquestions. If we wish to conduct a sequential analysis only on the effect of adult questions on child talk, we create a code translation file that includes the distinction between questions and nonquestions for adults.

A code translation file also allows the experimenter to direct STRAN to recode hierarchically organized codes. For example, our original coding distinguished various question types. STRAN can recode these as one category.

When running STRAN, the program "asks" if the experimenter wishes to eliminate "uncodables." The number and definition of speaker or bracketed codes considered uncodable can be changed within the system configuration file. For example, if the research question does not require unintelligible utterances to be considered a relevant event, then these may be eliminated from the STRAN output file.

The system configuration file also indicates the integer code that represents the "end of file" and "end of case." The end of file code is used to indicate the end of the transcript file to be analyzed. The end of case code is used to separate individual sessions when transcript files are pooled. The section on the POOL program (below) describes this procedure in more detail. The user can change the integers for the end of file and the end of case codes.

KAPPA. KAPPA compares two arrays of integer codes. We use it to compare two observer's judgments about the same transcript. The printout includes two types of tables: an agreement matrix and an agreement indices

table (see Appendix B). The input files for KAPPA are two vertical strings of integer codes with the same number of codes in each string. We use independent codings of the same transcript (i.e., output files of STRAN, which have been translated from the same translation file) as input files for KAPPA.

An agreement matrix is a two-dimensional matrix with the primary coder on the vertical axis and the comparison coder on the horizontal one. Agreements are tallied along the diagonal, and disagreements are tallied off the diagonal. With this matrix, an experimenter can identify systematic coder biases and specific categories that were coded in the case of a disagreement. In addition to total frequency, point-by-point agreement is used, because sequential analyses require sequential information to be reliable (Bakeman & Gottman, 1986).

The agreement indices table presents occurrence agreement frequency and percentage, nonoccurrence agreement frequency and percentage, total agreement frequency and percentage, and kappa (Cohen, 1960). Occurrence and nonoccurrence agreement are given because the absolute frequency of behavior types greatly influences the meaning of occurrence and nonoccurrence agreement (Nay, 1979). Kappas are reported to estimate interobserver agreement corrected for chance agreement, which is influenced by the base rate of the behavior.

LAGSEQ. LAGSEQ sequentially analyzes the output of the STRAN program and prints out descriptive and inferential statistics for all pairs of event types specified in the code translation file used to create the input file (i.e., for the corresponding STRAN run; see Appendix C). The experimenter specifies how many lags from the given event the program is to search for target events (range = 1-10).

For each event type, simple frequency, simple probability, and autocorrelation are printed out. Autocorrelation is the extent to which an event follows itself. Because the lag can be changed, autocorrelation can be calculated for each event or sequential pattern at lag 1 through lag 10. This is particularly important in conversations, because the turn-taking nature of conversations may result in the strongest pattern of autocorrelation being lag 2, rather than the traditional lag 1. Positive autocorrelation of the target and given behaviors can increase the Type I error rate. Negative values of the product of target and given behaviors' autocorrelation can increase Type II error rates (Gardner, Hartmann, & Mitchell, 1982). For more information on how autocorrelation may affect the absolute value and significance of chi-square-based tests of significance see Gardner et al. (1982) and Faraone and Dorfman (1987).

For each pair of event types, observed and expected sequential frequency, observed and expected transition probability, and the product of the target and given events' autocorrelation (i.e., joint autocorrelation) are printed. Four indices of the strength of the sequential dependency are printed for each pair of event types: (1) the z score (Allison & Liker, 1982), which is among the most common in-

dices for sequential dependency; (2) another form of z score, which is adjusted for the magnitude and direction of the product autocorrelation (i.e., Markov Z ; Faraone & Dorfman, 1987); this "corrected z " or Markov Z has been shown to be a robust statistic even when severe lag 1 autocorrelation is present (Faraone & Dorfman, 1987); (3) Pearson's r (Reynolds, 1984); and (4) Yule's Q (Bakeman, personal communication, October, 1988; Reynolds, 1984), which are computed to provide indices of sequential dependency that control for the base rates of target and given behaviors that are also independent of the number of events in the corpus.

Pearson's r and Yule's Q indices can be used as individual subjects' scores in parametric comparisons of average sequential dependencies between groups or conditions. Because this approach has not been used in the applied literature, we will discuss the need for and use of these indices in more detail here. An example of a research question for which comparing distributions of individual r s or Q s would be useful is the following: "Does the sequential dependency between adult questions and child replies differ in dyads with mentally retarded children versus those with nonretarded children?" The most frequently used indices of sequential dependency for such comparisons (i.e., obtained transition probabilities and z scores) lead to results that are difficult to interpret. It has been noted elsewhere that transition probabilities are influenced greatly by the base rate of the target behavior (Bakeman & Gottman, 1986). This makes using transition probabilities as individual scores in comparisons of sequential dependency between groups or conditions inappropriate because the base rate of the target behavior may systematically vary by group or condition. Z scores control for the base rate of the target and given behavior by comparing obtained sequential frequencies with expected ones (Bakeman & Gottman, 1986). However, even in transcripts with equal sequential dependency between and within target and given behaviors, z scores increase as the total number of events increase. Pearson's r and Yule's Q control for the base rate of the target behavior and are unaffected by the total number of events, provided no cell is equal to 0 (Reynolds, 1984).

We compared the corrected and uncorrected z scores, Pearson's r , and Yule's Q for two transcripts with equal sequential dependency but with different numbers of total utterances. The only meaningful difference between the two transcripts was that Transcript 2 had four times the number of utterances and pauses as did Transcript 1. This was accomplished by simply copying Transcript 1 onto itself three times to form Transcript 2. Table 2 presents this comparison.

In all cases, the z scores were twice as large in the longer transcript as they were in the shorter transcript. In contrast, the Pearson's r and the Yule's Q for the shorter transcript were identical with those of the larger transcript. Although Pearson's r and Yule's Q are not affected by

Table 2
Comparison of Various Indices
of Sequential Dependency on Two Transcripts
with Identical Sequential Dependency but Varying Lengths

Statistics for Child Talk/Adult Question	Transcript 1 ($N = 1,106$)	Transcript 2 ($N = 4,424$)
Descriptive Statistics		
Obtained transition probability	.68	.68
Expected transition probability	.03	.03
Joint autocorrelation	.01	.01
Indices of Sequential Dependency		
Uncorrected z	9.03	18.07
Corrected z	8.94	17.89
Yule's Q	.72	.72
Pearson's r	.27	.27

the total number of events, they are not corrected for autocorrelation. In addition, Pearson's r provides different information about the sequential dependency of interest from that which Yule's Q provides. Because Pearson's r is derived directly from the uncorrected z score (i.e., $r^2 = z^2$ of the 2×2 table / number of pairs in corpus), the Pearson's r represents a standardized form of the uncorrected z and is sensitive to marginal distribution. Yule's Q is insensitive to marginal distributions because it is derived from the cross-product ratio and consequently quantifies the sequential dependency quite differently than do the commonly used z scores and proposed Pearson's r . Therefore, the relative usefulness of the four indices of sequential dependency varies, depending on the data set, research question, and the experimenter's concept of categorical association.

LAGSEQ provides one final bit of information for each event pair. Using Bakeman and Gottman's (1986, p. 138) application of Siegel and Castellan's (1988) formula, the program indicates whether a particular input file has a sufficient expected probability and number of pairs to allow a reasonable test of the difference between expected and observed sequential frequencies. The formula is as follows: Number of pairs = $9/P(1-P)$, where P = expected transition probability. If an insufficient expected probability or number of pairs exists, then the printout indicates how many pairs of events are necessary to test the significance of a specific sequential dependency given the indicated expected probability.

POOL. If the individual transcripts meet the assumptions for such pooling (e.g., stationarity), they can be concatenated and analyzed as a pooled sample. The POOL program joins individual STRAN output files. In doing so, POOL inserts an end of case code that enables LAGSEQ to skip over the event windows including the last and the first code of adjoining individual STRAN output files (Bakeman, 1983). Pooled samples can then be sequentially analyzed as one transcript, with several breaks in the sequence corresponding to the points at which individual files are joined. To run POOL, the ex-

perimeter translates the transcripts to be joined by using STRAN and creates a list file of the file names of individual STRAN output files that are to be joined.

Example

We have included, in Appendix A, an example of a short coded transcript, code translation file, and the STRAN output file to help the reader follow the translation process. Examples of the tables from KAPPA (Appendix B) and LAGSEQ (Appendix C) are provided for a longer transcript because kappa and sequential analysis are most appropriate for data with many observations.

Availability and Documentation

The SATS system is available for dissemination to interested parties. This package includes (1) the source code and a documentation file for each of the programs, (2) an individual and a pooled data set, (3) the output of these data so that readers can verify that the programs are running correctly on their systems, (4) sample code translation and configuration files for the test data sets, and (5) support of the second author by phone for reasonable period while getting the program up and running. The package is available from the authors on request, at cost. Readers can get the package through BITNET or INTERNET or by mail on an IBM-compatible disk format. Simply contact Jon Tapp at (615) 322-8147 from 9:00 to 5:00 CST weekdays or on BITNET (TAPPJT@VUCTRVAX) or INTERNET (TAPPJT@ctrvax.Vanderbilt.Edu).

REFERENCES

- ALLISON, P. D., & LIKER, J. K. (1982). Analyzing sequential categorical data on dyadic interaction: A comment on Gottman. *Psychological Bulletin*, *2*, 292-403.
- BAKEMAN, R. (1983). Computing lag sequential statistics: The ELAG program. *Behavior Research Methods & Instrumentation*, *15*, 530-535.
- BAKEMAN, R., & GOTTMAN, J. M. (1986). *Observing interaction: An introduction to sequential analysis*. Cambridge, England: Cambridge University Press.
- COHEN, J. (1960). A coefficient of agreement for nominal scales. *Educational & Psychological Measurement*, *20*, 37-46.
- FARAONE, S. V., & DORFMAN, D. D. (1987). Lag sequential analysis: Robust statistical methods. *Psychological Bulletin*, *101*, 312-323.
- GARDNER, W., HARTMANN, D., & MITCHELL, C. (1982). The effects of serial dependence on the use of chi-square for analyzing sequential data in dyadic interactions. *Behavioral Assessment*, *4*, 75-82.
- MILLER, J., & CHAPMAN, R. (1983). *SALT: Systematic analysis of language transcripts*. Baltimore: University Park Press.
- NAY, W. R. (1979). *Multimethod clinical assessment*. New York: Gardner Press.
- REYNOLDS, H. T. (1984). *Analysis of nominal data* (2nd ed.). Beverly Hills, CA: Sage Publications.
- SIEGEL, S., & CASTELLAN, N. J., JR. (1988). *Nonparametric statistics for the behavioral sciences* (2nd ed.). New York: McGraw-Hill.

APPENDIX A

Example SALT transcript: Test.trans

C Cookie.
 A That Cookiemonster [tco] [com].
 A And Grouch [com] [tco].
 A Oscar the Grouch [tco] [com].
 C Oscar.
 A You wanna bus [tin] [ynq]?
 P (x).
 C Huh?
 P (x).
 A What's this called {phone} [tin] [whq]?
 A Telephone [tco] [com].
 C Phone.

Example Code Translation File: Test.cod

A[ynq],100,adult question
 A{whq},100,adult question
 A[com],200,adult comment
 C,300,child talk
 P,400,pause
 A,500,adult uncodable utterance

Example STRAN output file for Test.trans using Test.cod: Stran.output

300
 200
 200
 200
 300
 100
 400
 300
 400
 100
 200
 300
 9

APPENDIX B

Example of KAPPA Agreement Matrix

Kappa MATRIX output:

Rows are the primary dimension File: stran.output-py
 Columns are the comparison dimension File: stran.output-jt

916	400	100	300	200	500
400	261	0	0	0	0
100	0	487	0	0	0
300	0	0	111	0	0
200	0	3	0	44	0
500	0	0	0	0	10

APPENDIX B (Continued)

Example of Agreement Indices Table from KAPPA

Kappa statistical output:

CODE	OCCURRENCE		NON-OCCURRENCE		OVERALL		EXPECTED FREQUENCY
	NUMBER	RATIO	NUMBER	RATIO	NUMBER	RATIO	
400	261	0.996	654	0.998	915	0.999	74.65
100	487	0.994	426	0.993	913	0.997	260.51
300	111	1.00	805	1.00	916	1.000	13.45
200	44	0.936	869	0.997	913	0.997	2.26
500	7	1.000	909	1.000	916	1.000	0.05

AGRMT RATIO: 0.9956

KAPPA: 0.9929

APPENDIX C

Example Event Table from LAGSEQ:

Event Lag Sequential Output Listing

File of Events: stran.output

Output File Name: lagseq.output

Code File Name: code.file

Lag:1

Code	Simple Frequency	Simple Probability	Label
100	104	.0916	adult question
200	392	.3354	adult comment
300	393	.3463	child talk
400	243	.2141	pause
500	3	.0026	adult uncodable utterance

Example of Sequential Dependency Table from LAGSEQ:

Given event code: 100

Label for given event: adult question

Current Lag: 1

Autocorrelation for given event: -.0259

Target Event Code	Sequential Frequency	Transition Probability	Expected Sequential Frequency	Expected Transition Probability	z	Joint Auto- correlation	Corrected z	Yule's Q	Pearson's r	Analysis OK?	N Needed
100	7	.0673	9.5295	.0084	-.9021	.001	-.9015	-.1800	-.0268	Yes	
200	6	.0577	35.9189	.0316	-6.4738	.006	-6.4378	-.8144	-.1922	Yes	
300	74	.7115	36.0106	.0317	8.2151	.012	8.1205	.6937	.2440	Yes	
400	17	.1635	22.2661	.0196	-1.3208	-.007	-1.3302	-.1820	-.0392	Yes	
500	0	.0000	.2749	.0002	-.5508	.000	-.5508	na	-.0164	No	37169