ASSESSING CO-TERMINATION IN THE UNITIZING PHASE OF CONTENT ANALYSIS: A MULTI-RESPONSE RANDOMIZED BLOCKS PERMUTATION APPROACH

A Thesis

Presented in Partial Fulfillment of the Requirement for

The Degree Master of Arts in the

Graduate School of The Ohio State University

By

Li Cai, B. A.

* * * * *

The Ohio State University
2003

Master's Examination Committee:

Dr. Andrew F. Hayes, Adviser

Dr. William P. Eveland, Jr.

Approved by

Adviser

School of Journalism and Communication

ABSTRACT

The assessment of intercoder agreement in the unitizing phase of content analysis has long been overlooked. In particular, little attention has been paid to the issue of co-termination, which refers to the level of agreement among pairs of coders to break a given segment of content at the same points into smaller units. Although the kappa agreement coefficient can be used for the purpose of summarizing the agreement of co-termination, it has some inherent limitations. A new family of coefficients based on the Multi-response Randomized Blocks Permutation procedure is presented and numerical results are given.

To my family

ACKNOWLEDGMENTS

I am greatly indebted to my adviser, Dr. Andrew Hayes, for the encouragement and support that made this thesis possible, and for his patience in correcting my errors.

I also wish to thank my committee member, Dr. William Eveland, for the stimulating discussions that sparked some of the most important ideas of this thesis.

VITA

April 7, 1980 . . . . . Born – Nanjing, China

2001 . . . . . B.A. Business Communication, Nanjing University

2001 – present . . . . . Graduate Teaching and Research Associate
The Ohio State University

FIELDS OF STUDY

Major Field: Journalism and Communication

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Content analysis is a quantitative research methodology widely employed in the field of communication. Berelson's (1952) often cited definition of content analysis as an objective, systematic, and quantitative endeavor to describe the content of communication messages clearly endorsed the usefulness of content analysis to communication scholars. Given the importance of a certain research method, one would expect research reports using such a method to be well represented in communication journals, and such is indeed true for content analysis. In a recent "content analysis of content analyses" of articles published in *Journalism and Mass Communication Quarterly* between 1971 and 1995, Riffe and Freitag (1998) located 486 full-length reports using content analysis, comprising of roughly one fourth of the total number of articles published. They demonstrated an increasing trend of utilizing and featuring content analysis in communication research over the past quarter of a century, and they argued, as electronic databases and archives became more accessible to researchers and communication practitioners, this trend was likely to continue. An earlier study by Wilhoit (1984) suggested that more than 20 percent of theses and dissertations listed in *Journalism Abstracts* used content analysis. Moffett and Dominick (1987) reported a

similar result that 21 percent of the articles published in *Journal of Broadcasting* between 1970 and 1985 employed content analysis. Fowler (1986) confirmed the centrality of content analysis in the field of communication by showing that 84.1 percent of master's level research methods courses in journalism and communication graduate programs included content analysis. I also conducted an informal "content analysis of content analyses" published by *Journalism and Mass Communication Quarterly* between 1999 and 2001 and found that 43 papers used content analyses methodology, representing over 33% of all the research reports published in this journal.

As Krippendorff (1980) succinctly remarked, making "inferences from essentially verbal, symbolic, or communicative data" has always been at the heart of content analysis (p. 20). In order for scientific inferences to be valid, one must first ascertain the reliability of the research instrument. Just as chemists could ill-afford an uncalibrated balance in a chemical experiment, one could hardly imagine living a life in the communication scholarship without assessing the reliability of content analysis. Of course, if a stream of messages is to be analyzed by a well-designed computer program, one can probably worry less about reliability, but in most instances, if not all, content analyses still require much human labor, and thus the errors of human analysts become almost inevitable.

The standard process of content analysis, as described in most introductory communication research methods textbooks (e.g. Frey, Botan, & Kreps, 2000; Stewart, 2002; Wimmer & Dominick, 1994) essentially involves a *coding* process. Guetzkow (1950) observed that any transformation of qualitative data into a form "susceptible to quantitative treatment constitutes *coding*" (p. 47). He further emphasized that the coding

process could be broken down into two related phases: that of separating the qualitative material into units, and that of classifying the unitized data into established categories. The former is often termed *unitizing*, and the latter *categorizing*. The two processes are integral elements of content analysis yet require different strategies of reliability assessment. One term that needs a little clarification is *categorizing*. In many practical situations the coded units are indeed later classified into categorical sets, but this is not necessarily true. Coded units may be rated on ordinal, interval or ratio scales in subsequent coding procedures. The term *categorizing* will remain in use in the paragraphs to follow, but without any implication of merely categorizing the coded units into qualitative (nominal) sets.

In the coding process, usually a set of human *coders* or *judges* are involved. The assessment of reliability of the content analysis thus becomes an assessment of the reliability of the coders, even though this is not a sufficient condition for the entire content analysis study to be reliable, the coding process is of such importance that low intercoder reliability would render all subsequent analyses meaningless, because low intercoder reliability would suggest that the obtained results were largely not replicable (Krippendorff, 1980, p. 131). Ideally, the coders should be trained to rate or judge the content independently and yet to arrive at the same ratings in precisely in the same manner as intended by the coding scheme.

Intercoder reliability is established when the same pieces (possibly a very large number) of content yield same ratings from independent coders using a common data language (Krippendorff, 1980, p. 133). Formally, the term intercoder reliability should be more appropriately termed *intercoder agreement* (cf. Lombard, Snyder-Duch, &

Bracken, 2002), but the two terms will nevertheless be used interchangeably in the pages to follow since given the present context the meaning of the two terms is not much different. Another distinction that should be made is the "depth" at which the messages are to be coded. Berelson (1952) clearly intended content analysts to deal only with the manifest content, i.e. the information "as is," without invoking additional mental efforts of the coders to discover the latent content or the implied meaning. However, unless the research question can be easily answered by simply counting the number of words in a newspaper article or the number of occurrences of the names of candidates in pre-election news coverage – which can be quite easily done with a computer – the coding process will often require the coders to make subjective judgments. Under those circumstances, readers of the research would demand the researchers to demonstrate that "those judgments, while subjectively derived, are shared across coders," which again confirmed the practical necessity of establishing intercoder agreement in content analysis (Potter & Levine-Donnerstein, 1999, p. 266).

Having illustrated the importance of intercoder agreement, the current status of correctly using and reporting intercoder agreement measures in communication journals is quite alarming. Riffe and Freitag (1997) found that only half of the 486 articles published in *Journalism and Mass Communication Quarterly* between 1971-1995 reported intercoder reliability. A recent study by Lombard et al. (2002) searched virtually all content analysis articles indexed in *Communication Abstracts* from 1994 to 1998 and of the 200 articles they found, only 69% ever mentioned intercoder reliability, and usually the methods for computing intercoder reliability were not reported. Of the 44% of all articles that did report the names of the specific methods, more than half of

them relied on liberal indices that are not chance-corrected, such as percent agreement, which seriously undermined the effort of computing and reporting reliability coefficients. I found that of the 43 content analyses published in *Journalism and Mass Communication Quarterly* from 1999 to 2001, only 65% reported using some form of intercoder reliability measurement. More than one third of the articles that reported reliability coefficients were still using Holsti's (1969) method, which is not chance-corrected. In addition, two papers employed Pearson's correlation coefficient *r* when assessing the agreement of categorical coding, which is also a very poor practice that can hide huge proportions of disagreement.

Given the current undesirable state of affairs of appropriately using and reporting intercoder agreement indices in the communication scholarship, the next section shall explicate intercoder agreement in the context of a two-stage coding process, namely, unitizing and categorizing. The importance of *co-termination* when unitizing textual data shall also be presented.

CHAPTER 2

CO-TERMINATION

When Guetzkow (1950) wrote about unitizing and categorizing, he presented a convincing case that in order for the entire content analysis to be reliable, one has to regard the assessment of the overall intercoder agreement as a two-stage process. Ideally one should compute agreement measures for unitizing first and then calculate the agreement indices for categorizing, with the "overall" reliability referring to the combined intercoder agreement in both stages. One should note that this overall reliability does not always have to be expressed in quantitative terms. It is possible that a particular content analysis consists merely of categorizing existing units, and then this two-stage notion would not be relevant. However, there are times when unitizing is a must, and under such circumstances, the intercoder agreement of unitizing becomes crucial. This paper does not attempt to develop any new agreement indices for the categorizing phase, as there are established methods already. Instead, the aim is on how the agreement of unitizing can be better summarized, and this goal cannot be achieved without first understanding the complexities of intercoder agreement in the unitizing phase.

The agreement of unitizing focuses on how independent coders choose breaking points at various places in a *segment* of content, be it a sentence, a paragraph, an article, or an entire television show. The segments are assumed to be clearly delineated from one another and are usually naturally given. This assumption is not unfair because most of the qualitative content that can serve as segments for coding has unambiguous endpoints. For example, if a newspaper article is chosen as a segment, where it ends is crystal clear. It is further assumed that segments are of two types, discrete and continuous.

Discrete segments are composed of a finite number of *elements*. Defining what is an *element* is very difficult, and in a sense it is an inherent flaw of discrete segments, because 1) it depends on the research question, 2) how detailed the researcher would like the content analysis to be, and 3) for two researchers using same piece of content segment, if the element is defined in different ways, the two will necessarily come up with different intercoder reliability coefficients. But for now, examples should at least help illustrate what constitutes an element. Consider, for instance, a sentence from an online chat transcript: "Apparently, from what I read, they haven't identified the dead body yet." It is convenient to define a word – anything in between two spaces – as an element, and this is a segment containing 12 elements. Thus defined, unitizing becomes an operation of grouping elements into units. Generally speaking, a *unit* is a subset (not necessarily a strict subset) of a segment. In the context of discrete content, a unit may contain one or more elements, and a segment may contain one or more units. As a concrete example, Schaefer (1999) analyzed news reports of the State of the Union Address in the New York Times at the *assertion*-level, and according to his definition, a paragraph in an article may contain multiple assertions, so it is easy to infer that a sentence, in this case,

can serve as an element. Intercoder disagreement arises when coders define the units differently. Suppose that a paragraph contains 5 sentences, and there are two judges coding this paragraph. The first coder grouped the first two sentences into an assertion while the other grouped the first three into one assertion. It is easy to see that they are in disagreement.

The idea of an *element* is not applicable to continuous segments. For instance, a researcher may want to unitize audio/video recordings. It is probably hard to define what an element is within a continuous stream of audio/ video recording, but it is easy to deal with the relative *length* of a unit, perhaps expressed in terms of time. One can imagine the coder using a stopwatch to record the lengths of units, and intercoder disagreement occurs when the coders come up with different length readings. For instance, suppose two coders are involved in unitizing a 10-minute segment of audio recording into 2 units, and one of the coders defined the first unit to be 5 minutes long, while the other defined it to be 6 minutes long. Apparently the two coders are in disagreement for this continuous segment.

The idea of length is widely applicable and one can essentially re-express the discrete type of unitizing using lengths readings as well. The basic idea is to define the length of a discrete unit as the number of elements it contains. Consider this example: ABCD, a discrete segment of 4 elements, is to be coded by two judges by putting slashes at the breaking points. Judge 1 gave: A/B/CD, and judge 2 gave: A/BC/D. They both came up with three units for this segment, and the reliability data, using discrete terms, is a set of binary streams: 1 1 0, for judge 1; and 1 0 1 for judge 2. The number of entries in the binary streams is the number of elements minus 1, representing the maximum number

of possible breaking points. In this case, there are altogether 3 possible breaking points, between A and B, between B and C, and between C and D. The 1's in a stream signify observed breaking points. For example, the first coder broke between A and B, which is the first possible breaking point in the segment, so the first entry in the binary stream is 1. Using the same logic, since the second coder did not break between B and C, the second entry in the binary stream corresponding to this coder is a zero. The same data can be expressed in terms of lengths: 1 1 2 for judge 1; and 1 2 1 for judge 2. The numbers correspond to the number of elements in a particular unit, and the total number of entries equals the number of units. For instance, the first unit for judge 1 contains one element – "A," therefore the corresponding length reading is 1. Take the second unit for judge 2 as another example. The length is 2 because "BC" contains 2 elements.

Having defined the terms, it is natural to introduce the concept of *co-termination* and review what Guetzkow (1950) recognized as the two kinds of errors of unitizing a stream of content: (1) failure to agree on the breaking points between the units, and (2) failure to attain the same number of units (p. 54). Co-termination, or co-terminability, a term introduced but not clearly defined in Guetzekow (1950), refers to the agreement among pairs of coders to break a given segment of content at the same points into the same number of smaller units. Note that this definition essentially contains two components: (1) the agreement on the breaking points, and (2) the agreement on the number of units. Such a definition of co-termination is said to be in a strong form because there will be perfect agreement of unitizing among coders when the strong form of co-termination is achieved. It is the necessary and sufficient condition for a weaker form to exist because, for example, it is possible that a pair of coders agree partially on

how to choose the breaking points and yet at the same time do not agree on how many units there are in the segment of content. Suppose two coders were instructed to break an article into smaller units containing one or more paragraphs. The two coders started out in perfect agreement as to how to group the paragraphs into units up to a certain paragraph after which things started to fall apart. As a result, the numbers of units were different, and certainly by definition of strong co-termination, they failed to achieve agreement. However, one has to acknowledge that at least the two agreed somewhat in the beginning, and a good agreement measure should give partial credit to such agreement. It is conceivable that any measure of agreement based on the strong form of co-termination would necessarily be a conservative one and thus the existence of a weak form of co-termination is not an idea plucked out from the thin air.

The weak form of co-termination essentially depends on the sequential nature of content streams, i.e. one can only start unitizing from the beginning of a segment and proceed as the stream goes. Of course, going backwards from the end is not impossible, but this is makes little difference because one can then define the end as the beginning. Expressed in discrete terms, the weak form of co-termination between two-coders is defined as choosing breaking points so that at least the two coders grouped one set of elements in the same manner. Consider the previous example again: a segment – ABCD, with 4 elements, and 3 coders were to unitize it. The result happened to be as follows: coder 1 – A/B/CD, coder 2 – AB/C/D, and coder 3 – A/B/C/D. There are three distinct pairs of coders: 1 vs. 2, 2 vs. 3, and 1 vs. 3. Clearly, none of the pairs achieved co-termination if the strong definition is used. Coders 1 and 2 gave the same number of units but were not co-terminus. Although coders 1 and 3 gave different numbers of units

(3 and 4, respectively), they actually attained the weak form of co-termination for the groupings of A and B into the first and second unit. For coders 2 and 3, they achieved co-termination for C and D. The basic conceptualization of the measurement of co-termination would be to employ the strong definition when the coders agree on the number of units and to use the weak form when the numbers of units are different. Henceforth, it shall be implied that the strong definition is used whenever the numbers of units are the same; otherwise, the weak definition will be utilized.

It is worthy of pointing out that according to Hubert (1977) there are three definitions of agreement when the number of coders goes beyond two: DeMoivre's definition, target-rater definition, and pair-wise definition. The first one refers to the unanimous agreement of all coders, and the second one refers to the joint agreement of all other coders with a "target-rater" who provides the "true" rating, and the third, which is also what is implied in the definition of co-termination, refers to the agreement between any pairings of coders. It is easy to see that DeMoivre's definition tends to yield the most conservativeness. Most of the popular intercoder agreement indices that can handle three or more coders use the pair-wise definition, as does the new coefficient to be proposed in subsequent sections.

Having defined what co-termination is, it is not difficult to infer that the mere agreement on number of units does not imply co-termination. As to the relative importance of the two, Guetzkow (1950) remarked that the failure to achieve "co-terminability" is less likely to lead to confusions and low intercoder reliability in the subsequent categorizing of the coded units (p. 55). There is absolutely nothing wrong with this argument, because how far reliability assessment should go is a practical matter

related to the nature of the specific study at hand. If the unit boundaries are relatively clear, or if slight inconsistencies in co-termination do not significantly affect the subsequent use of the coded units, one could worry less about co-termination and focus more on achieving a high level of agreement on the number of units. However, there are certain times when disagreement in co-termination may lead to different interpretations of the same data, even though the number of units are the same across coders. For instance, if two coders were to divide the sentence "Apparently, from what I read, they haven't identified the dead body yet," and the coders agreed that it contained two units, but the first coder put the division mark right after "apparently," while the second put it after "read." The interpretation of the two units would necessarily be different, because a stand-alone "apparently" would suggest confirmation, while "apparently, from what I read" would refer to the clear inferences that the chat user could make from what he or she read. This example might be a very trivial one. What is important is to realize that the mere agreement on number of units does not automatically imply reliability of unitizing. Still using the previous example, suppose that the first coder divided the sentence after both "apparently" and "read," and the second coder only divided the sentence after "apparently," the number of units for the two coders are 3 and 2, respectively, and there seems to be much disagreement between the two, but in fact they did achieve co-termination, at least for the first unit. Given such results, at least the interpretation for the first unit – "apparently," is unambiguous.

CHAPTER 3

POPULAR INDICES OF INTERCODER AGREEMENT

This section briefly examines the five most widely used intercoder reliability indices in the communication literature and explicates their limited applicability to the measurement of co-termination in the unitizing phase of content analysis. Most of them are intended for bivariate nominal level coding. For discrete unitizing reliability data (binary streams) between two coders, Cohen's κ can be used, but only to a limited extent. For continuous content, no current indices are directly applicable.

Percent Agreement and Holsti's Method

This is perhaps the most easily understood method for calculating intercoder agreement for the categorizing phase. It is simply the "percentage of all coding decisions made by pairs of coders on which the coders agree" (Lombard, et al., 2002, p. 590). This is not a chance corrected measure, and Krippendorff (1980) illustrated how chance could artificially inflate percent agreement with a neat example (pp. 133-135). In general, using percent agreement is a very poor practice that can artificially inflate agreement.

Holsti (1969) proposed a variation of the percent agreement measure that does not require the two coders to be coding the same pieces of content. His formula can be expressed as

$$\text{Agreement} = \frac{2N}{N_1 + N_2},$$

where N is the total number of coding decisions the two coders agreed upon, and N_1 and N_2 are the numbers of coding decisions by the first and the second coder, respectively. When two coders are coding the same pieces of content, this formula is the same as percent agreement. This is still not a chance-corrected measure and it suffers from the same drawbacks as percent agreement. Even though some prominent statisticians have argued against the use of chance-corrected measures (e.g., Goodman & Kruskal, 1954), supporters of chance-corrected measures "far outweigh detractors" (Berry & Mielke, 1988, p. 922).

Scott's π

This is a chance-corrected index first introduced by Scott (1955) primarily in the context of coding qualitative data obtained from surveys. In its original form, this index is only applicable to nominal level coding and accommodates only two coders, although it is worth mentioning that Craig (1981) has given an extension of Scott's π to the case of multiple coders. Scott's π's basic formulation is the ratio $(P_o - P_e)/(1 - P_e)$, where P_o is the proportion of observed agreement, and P_e is the proportion of agreement expected by chance. Usually it is assumed that two coders independently classify each of the n units into one of c established categories. The layout for computing π essentially involves the construction of a two-way cross-classification table, with entries in the table being the proportion of observations falling into one of the c-by-c cross-classifications. Scott's π is the first coefficient that considers the expected agreement as a function of both the number of categories and the marginal distributions, but its assumptions are over

simplifications of the reality. The π coefficient assumes that the column and row marginal distributions are identical to the "true" proportions, and that the two coders share the same marginal distributions. In other words, when P_o is unambiguously given by the sum of the diagonal elements of the *c*-by-c cross-classification table, P_e is taken to be the sum of the squares of "true" marginal proportions. Given the context of survey research, where π originated, the former assumption is not unreasonable, as the "true" proportions are usually obtainable, and in some situations this assumption has given Scott's π a distinct edge over similar coefficients like Cohen's κ, because π can still be computed when the two coders have coded different pieces of the content, while computation of κ requires that the pair of coders have coded the same units (Craig, 1981, p. 261). However, it is the latter assumption of π that is particularly problematic. As Cohen (1960) pointed out, "one source of disagreement between a pair of judges is precisely their proclivity to distribute their judgments differently over the categories" (p. 41). Furthermore, the "true" proportions are not always available outside of the field of survey research. Making such unrealistic assumptions only hinders the usefulness of the π coefficient. Within communication, however, π is perhaps still the most widely used measure of intercoder agreement. In the informal content analysis of research reports in *Journalism and Mass Communication Quarterly*, I found 28 papers that reported some form of reliability assessment. 10 of them used π, which is as popular as the Holsti's (1969) method.

Cohen's κ

Cohen's (1960) κ is defined in much the same way as Scott's π, in that both coefficients require the construction of a *c*-by-*c* cross-classification table to calculate the

agreement index. The κ coefficient, and its variants for bivariate nominal level coding takes the familiar form of a ratio between observed and expected proportions, $\kappa = (P_o - P_e)/(1 - P_e)$, with P_o given by the sum of the diagonal elements of the *c* by *c* cross-classification table, and P_e is found by first multiplying each column marginal with its associated row marginal and then taking the sum of the products. Note that the P_e's are calculated differently for κ and π, and for bivariate nominal level coding, this is the only difference between the two coefficients. One can easily see that κ takes into account the difference in the two coders' marginal distributions when calculating the expected agreement.

Cohen's κ has enjoyed continued development by psychological methodologists. Cohen (1968) himself introduced a weighting procedure that accounts for the differential severity of disagreements. Fleiss (1971) gave its extensions to the case of multiple raters. Fleiss and Cohen (1973) established the equivalence of weighted κ and the intra-class correlation coefficient. Hubert (1977) introduced the underlying mathematical model of matching distributions in probability theory to users of the κ coefficient. Fleiss, Nee, and Landis (1979) worked out κ's asymptotic variance. Conger (1985) extended it to measure agreement over time for continuous nominal scales. However, the κ coefficient is not as popular in communication as it is in other social sciences such as psychology.

Krippendorff's α

When the number of coders is exactly two with nominal level coding assumed, Krippendorff's (1970) α coefficient is identical to Scott's π (Krippendorff, 1980, p. 138). What makes the α coefficient more appealing than its competitors is that it offers an easy

extension to measure the agreement of higher levels of measurement and of multiple coders. Recall that Guetzkow (1950) described the two kinds of errors in unitizing textual data. It appears that Krippendorff's α coefficient may well serve the purpose of calculating the intercoder agreement of the number of units of a given segment of content. Krippendorff's α is not very widely used by communication researchers either. I found in the "content analysis of content analyses" of the 28 *articles in Journalism and Mass Communication Quarterly* between 1999 and 2001, only three used Krippendorff's (1970) α and three used Cohen's (1965) κ, as compared to the ten articles using Holsti's (1969) method.

CHAPTER 4

MULTI-RESPONSE RANDOMIZED BLOCKS LAYOUT AND INTERCODER AGREEMENT

This section describes some of the details of the Multi-response Randomized Blocks Permutation procedure (MRBP) relevant to the assessment of agreement. MRBP is a variation of the Multi-response Permutation Procedure (MRPP) (Mielke, Berry, & Johnson, 1976). It is first introduced by Mielke & Iyer (1982) as a supplement to MRPP. Both MRPP and MRBP are based on the general principle of permutation tests (for details of MRPP, please refer to Appendix A; for an in depth treatment of permutation tests, consult Edgington, 1987). In its original formulation, MRBP defines a *b*-block by *g*-treatment randomized blocks experiment and within each block there is only one *r*-dimensional observation per treatment, taken as $n = 1$ for each cell. The reader can think of the MRBP layout as a *b*-row by *g*-column table, and of course, there are altogether (*bg*) cells in this table. In each cell, there is only one observation. This observation can be a multivariate / multidimensional response vector ($r > 1$) or it can be a scalar ($r = 1$). MRBP makes use of the distances between these multidimensional response vectors when constructing the test statistic.

Let $\mathbf{x}_I' = (x_{1I}, \ldots, x_{rI})$, and $\mathbf{x}_J' = (x_{1J}, \ldots, x_{rJ})$ be the transposes of two r-by-1 vectors of multivariate responses. The symmetric distance function between the two multidimensional response vectors – $\mathbf{x}_I$ and $\mathbf{x}_J$ – is given by

$$\Delta_{I,J} = \left[\sum_{c=1}^{r} (x_{cI} - x_{cJ})^2 \right]^{v/2}, \qquad (1)$$

where x_{cI} and x_{cJ} are the corresponding elements in the r-dimensional response vectors. It is easy to see that the distance between two multivariate responses is a power function of the sum of the squared differences between each element and therefore the choice v gives rise to a variety of distance functions. The value of v determines the analysis space of the test and choice is somewhat arbitrary, but the most widely used two are $v = 1$ and $v = 2$, which corresponds to metric Euclidean space (the triangle inequality holds) and non-metric squared Euclidean space (the triangle inequality fails). Some of the most widely employed tests such as the *t*-test, *ANOVA*, and their multivariate counterparts – Hotelling's generalization of Student's *t*, and Bartlett-Nanda-Pillai trace test in *MANOVA* all use squared Euclidean analysis space. Berry and Mielke (1988) pointed out that the choice of squaring the distances is "questionable at the best" (p. 922). They suggested $v = 1$ be used at all times, but Janson and Olsson's (2001) modified agreement statistic uses $v = 2$ and their main argument for the more conventional metric is the ease of interpretation. As the reader will see later, for the binary streams of intercoder agreement data when unitizing discrete content, the choice of v does not matter, but for continuous content, $v = 2$ is sometimes the only choice due to the vast simplification of the mathematics.

Using the MRBP layout, Berry and Mielke (1988) provided a re-formulation of Cohen's κ for the categorization phase of content analysis and a natural extension of κ to multiple coders, and to higher levels of measurement. In brevity, the original cross-classification layout of κ is transformed into a b-block by g-treatment MRBP layout. For example, assuming that two observers independently coded each of the g units into one of the r categories, the familiar cross-classification layout of κ would be an r by r table with the entries in the table being the proportions of cross-classifications in particular cells. The MRBP layout, on the other end, would be a 2-block by g-treatment table with a total number of $(2g)$ r-dimensional response vectors in each one of the $(2g)$ cells of the table. The number of coders corresponds to the number of blocks, and the number of treatments, or in other words – the number of columns, represents the number of units categorized or equivalently, the number of coding decisions made. Suppose that the number of categories – r equals 3, then in this case, the response vectors would all be 3-by-1 in dimension. If the first coder classified the first unit into the first category, the response vector associated with that coding should be stored in the cell at the intersection of the first column (or equivalently, treatment or unit) and the first row (or equivalently, block or coder), and it would take the form of $(2^{-½}\ \ 0\ \ 0)'$. If the second coder classified this unit into the second category, the response vector should be entered in the cell corresponding to the first treatment of the second block (meaning, the first unit of the second coder), and this vector takes the form of $(0\ \ 2^{-½}\ \ 0)'$. Generally speaking, if a coder classified a unit as belong to the ith category, the ith element in the response vector would be $2^{-½}$, and all other elements would be 0. The relative location of $2^{-½}$ in a response vector indicates the category into which the coder has assigned the particular

unit. And the choice of using $2^{-½}$, not just any other number, is to ensure the nominal level property of κ, i.e. the distance between the two vectors will be zero if the two coders agree, and one if the two disagree.

The extended measure of agreement is given by the equation

$$\kappa = 1 - \delta_{obs} / \mu_{\delta}, \qquad (2)$$

where δ_{obs} denotes observed disagreement and μ_{δ} denotes expected proportion of disagreement by chance. Because MRBP is a based on permutation, μ_{δ} is found by permuting the data within each block across treatments. In other words, μ_{δ} is found by permuting the data from each coder across units. In the original experimental design context, the definition of μ_{δ} reflects the addition of blocks because as a general principle in randomized blocks designs, from which MRBP originated, randomization does not occur across blocks, and therefore in constructing a permutation test, data cannot be permuted across the blocks. This is also implied by the matching distribution in elementary probability theory, which forms the underlying probabilistic model of κ (see Hubert, 1977). The maximum number of permutations is $M = (g!)^b$ – the total number of permutations within each block, or stated equivalently, for each coder's responses, to the bth power. Such a formulation makes the extension of κ to multiple coder situations very easy.

Generally, assuming b coders independently categorized g units of content, let x_{kip} denote the elements in an r-by-1 response vector from coder i for unit p, where $k = 1, \ldots, r$, $i = 1, \ldots, b$, and $p = 1, \ldots, g$, the disagreement (distance) function between coder i and coder j for the pth unit is given by

$$\Delta_{ip,jp} = \left[\sum_{c=1}^{r} (x_{cip} - x_{cjp})^2 \right]^{v/2}, \tag{3}$$

and the observed disagreement over all distinct pairs of coders and over all g units is given by

$$\delta_{obs} = \left[g\binom{b}{2} \right]^{-1} \sum_{p=1}^{g} \sum_{i<j} \Delta_{ip,jp}, \tag{4}$$

where $i < j$ denotes the sum over all i and j such that $1 \le i < j \le b$, and basically this is to ensure that the response from a coder is not compared with itself.

Assuming that the M permutations are equally probable, a theoretical definition of chance disagreement is given by

$$\mu_\delta = M^{-1} \sum_{i=1}^{M} \delta_i, \tag{5}$$

However, one does not need to enumerate all M permutations to arrive at μ_δ, a more efficient working formula for μ_δ is available due to the fact that the first moment of the permutation distribution is a constant multiple of g^2 elementary calculations (see Mielke & Iyer, 1982).

Using similar notations as in equations (3) and (4), then the distance function between the ith coder for the pth unit and the jth coder for the qth unit is given by

$$\Delta_{ip,jq} = \left[\sum_{c=1}^{r} (x_{cip} - x_{cjq})^2 \right]^{v/2}, \tag{6}$$

and the following equation determines the chance disagreement

$$\mu_{\delta} = \left[g^2 \binom{b}{2} \right]^{-1} \sum_{p=1}^{g} \sum_{q=1}^{g} \sum_{i<j} \Delta_{ip,jq} \,, \tag{7}$$

where $i < j$ denotes the sum over all i and j such that $1 \leq i < j \leq b$. Equation (7) seems to be quite complicated. However, it is nothing but the average distance between any distinct pairings of response vectors. As Hubert (1977) suggested, this is also an existing result in the matching distribution literature. Berry and Mielke (1988) have named their extended κ coefficient as R, and have established the equivalence of this statistic with other known measures.

CHAPTER 5

THE PROPOSED FAMILY OF COEFFICIENTS OF CO-TERMINATION

MRBP Reformulated κ

It is assumed that two coders are present, and that they have broken a 7-word sentence into 3 units. The analysis of intercoder agreement, using discrete terms, may be expressed as a 2 block by 6 treatment MRBP table. Recall that the number of coders is equal to the number of blocks, and the number of coding decisions made is equal to the number of treatments (columns) in the MRBP table. If this sentence is of the form ABCDEFG, where each letter represents a word, decisions of choosing between "0" – not to break and "1" – to break, at possible breaking points (the spaces between two consecutive words) are repeated by each coder for six times. Therefore, the entries in the MRBP table are just 0's and 1's, and the data is summarized in Table 1.

The actual codings in Table 1 are: A/BCDE/FG for coder 1, and A/BCDEF/G for coder 2. If the usual cross-classification layout of κ is used, the design should be a 2 by 2 table, and it would look like Table 2. The diagonal entries represent the agreement between the two coders in choosing the breaking points (cell 1-1) or non-breaking points (cell 0-0), and the off-diagonal entries are their disagreement.

Cohen's κ can be calculated from Table 2 using the usual way.

$$\kappa = \frac{P_o - P_e}{1 - P_e} = \frac{(1/6 + 1/2) - [(1/3 \times 1/3) + (2/3 \times 2/3)]}{1 - [(1/3 \times 1/3) + (2/3 \times 2/3)]} = \frac{2/3 - 5/9}{1 - 5/9} = .25$$

For the computation of κ using MRBP layout, let x_{ip} represent the value (0 or 1) from the cell corresponding to the ith coder and pth possible breaking point, where i = 1, …, b, and p = 1, …, g, the symmetrical MRBP distance function – equation (6) – between any two cells x_{ip} and x_{jq} in a table similar to Table 1 can be reduced to

$$\Delta_{ip,jq} = \left[(x_{ip} - x_{jq})^2\right]^{v/2}, \tag{8}$$

note that the choice of v does not matter here because the values in the cells are either 0 or 1.

Using equations (2) – (8), a reformulated κ can be expressed as one minus the ratio between δ_{obs} – the observed disagreement – and μ_δ – the expected disagreement. μ_δ can be found by averaging over all δ's obtained from permuting data within blocks. Table 3 is an example of a possible permutation. For instance, in Block 1, the 1s originally in the 1st and 5th treatments are swapped into the 2nd and 3rd places. For this permutation δ = 4/6 = 2/3.

To summarize, δ_{obs} can be calculated using equation (4) as $\delta_{obs} = 6^{-1}\sum_{p=1}^{6}\Delta_{1p,2p}$ = (0+0+0+0+1+1)/6 = .333, and $\mu_\delta = 36^{-1}\sum_{p=1}^{6}\sum_{q=1}^{6}\Delta_{1p,2q}$ = 16 / 36 = .444, and κ is 1 – .333/ .444 = .25, which is exactly the same as using the cross-classification table, but from equations (2) – (8), one can see that the MRBP approach is inherently multivariate and can be readily extended to multiple coders unitizing the same segment of content. A

GAUSS (Aptech Systems Inc., 1997) procedure that implements formulae (2)-(8) is in Appendix B.

One of the problems with this approach, as the reader probably has already noticed, is an underestimate of reliability. Without calculating any statistics, a visual examination of the codings: A/BCDE/FG for coder 1, and A/BCDEF/G for coder 2, reveal the fact that the codings are not much different yet the agreement measure indicates that it is merely 25% agreement above chance, a value too low by any standards. In the next part of this paper, a small simulation study shall be presented to demonstrate the conservativeness of κ, but for now, a remedy shall be presented in the next section and it makes use of the notion of continuous content.

A second problem of significance to the use of κ for discrete type of content is the choice of the underlying basic *element*. This is especially pertinent when the content is not inherently discrete, and one wishes to transform the length readings to binary streams and use κ to calculate reliability.

Restricted Permutation and A Measure Based on Unit Lengths

One can re-express the data in Table 1 using unit lengths (defined as the number of elements in the units) and the result is summarized in Table 4. For instance, the "1" in the first cell of Table 1, which means an observed break at the first possible breaking point, translates into a unit of length 1. This is the first unit given by coder 1, and therefore it is entered into the first cell of Table 4. As another example, the "1" in the fifth column of Table 1 indicates another break by coder 1, and the 4 elements between this break and the first break form unit 2. This is apparently a unit of length 4. As a consequence, in Table 4 the second entry for coder 1 is 4. For the moment, the reader is

asked to ignore the lines corresponding to the "Cumulative" lengths. The usefulness of these identities will become clear later.

By simply applying equations (2) – (7) on the two coders, one can obtain the reliability coefficient fairly easily in this case. With $v = 1$, $\kappa = 1 - \delta_{obs} / \mu_{\delta} = 1 - .667 / 1.778 = .625$. With $v = 2$, $\kappa = 1 - \delta_{obs} / \mu_{\delta} = 1 - .666 / 5.111 = .87$. Using length readings, the agreement index increased quite a bit. However, the direct application of MRBP is quite problematic given that coders usually do not agree on the number of units either. Not only are the computational formulae thus rendered useless, there are conceptual problems as well.

Consider the coding, as summarized in Table 5: The first coder came up with 7 units for this continuous piece of content and the second one came up with only 6. One can easily think of replacing the one missing cell in the last column with zero and then apply the computational formulae, but a theoretical problem arises because when the permutation within the second block is conducted, the imputed zero may appear, for example, in the 3rd column. It makes little sense because one can hardly imagine a unit of length zero in between two other units of positive lengths. After all, the communication content is a sequential stream that does not stop until the endpoint.

This problem can be offset by using a restricted permutation approach, i.e., by fixing the trailing zero(s), should there be one or more missing cells in the last a few columns when conducting the within block permutations. Therefore, the total number of possible permutations in the given example is only (7!)(6!) = 3,628,800, instead of $(7!)^2$ = 25,401,600, as the zero x_{27} will remain un-permuted. More generally, the total number of permutations is given by

$$M = \prod_{j=1}^{b} g_j!, \tag{9}$$

and when all g_j's are equal, equation (9) is the same as $(g!)^b$, where g refers to the maximum number of units given by one or more coder(s) for a particular segment. This change from the original MRBP approach essentially reflects the usefulness of the so-called *reference subsets* described in Edgington (1987). If the set of $(g!)^b$ data permutations is taken as the primary reference set, then the computation of μ_δ under the condition when all g_js are equal would be using the reference distribution for the general-null hypothesis, whereas when not all g_js are equal, and thus equation (9) yields a smaller value than $(g!)^b$, the computation of μ_δ would be comparable to the test of a restricted null hypothesis (see Edgington, 1987, pp. 305-316).

When not all g_js are equal, the direct expression of μ_δ is quite cumbersome, thus it is useful to introduce the following computational expressions for clarity and computer implementation. A GAUSS (Aptech Systems Inc., 1997) procedure that implements these formulae is in Appendix C. Let (x_{ip}) denote a row of g_i unit length readings from coder i, e.g. a row in Table 5, where $p = 1, \ldots g_i$, let M be given as in equation (9), and let the between cell distance – $\Delta_{ip,jq}$ be defined as in equation (8), then for any choice of v, define the following equations:

$$C_1(i,j) = \frac{M}{\max(g_{i,}g_j)}, \tag{10}$$

$$C_2(i,j) = M\left[1 - \frac{\min(g_{i,}g_j)}{\max(g_{i,}g_j)}\right], \tag{11}$$

$$\psi(i,j) = \begin{cases} i & \text{if } g_i > g_j, \\ j & \text{if } g_i < g_j. \end{cases} \tag{12}$$

$$D_1(i,j) = \sum_{p=1}^{g_i} \sum_{q=1}^{g_j} \Delta_{ip,jq} \ , \tag{13}$$

$$D_2(i,j) = \sum_{p=1}^{g_{\psi(i,j)}} (x_{\psi(i,j)p} - 0)^v = \sum_{p=1}^{g_{\psi(i,j)}} x^v_{\psi(i,j)p} \ , \tag{14}$$

$$E(i,j) = C_1(i,j)D_1(i,j) + C_2(i,j)D_2(i,j) \, , \tag{15}$$

$$\mu_\delta = \left[M \binom{b}{2} \right]^{-1} \sum_{i<j} E(i,j) \, , \tag{16}$$

where $i < j$ denotes the summation over all i and j such that $1 \le i < j \le b$. Given the considerable modification to Mielke and Berry's (1988) original reformulation of κ, it is tempting to give this new coefficient a different name, say, κ'. For the data in Table 5, with $v = 1$, $\kappa' = 1 - \delta_{obs} / \mu_\delta = 1 - 3.486/ 5.265 = .338$. With $v = 2$, $\kappa' = 1 - \delta_{obs} / \mu_\delta = 1 - 23.811/ 50.174 = .525$. However, this restricted permutation approach is not without problems of its own. The major drawback is that the use of lengths sometimes fails to capture the hidden disagreements.

Consider an example (Table 6) in which two coders are supposed to unitize an audio clip by writing down the length of each unit – expressed in minutes and seconds – presumably using a timing device such as a stopwatch. Unlike the previous example in Table 4, where the use of length entails a transformation of the binary streams into length readings, these length readings are truly "continuous." One can easily see from the table that the total length of the content segment is 2 minutes, and both coders unitized the content segment into three units. The numbers in the parentheses are the length readings

expressed in terms of seconds. A direct application of equations (8) – (16) is possible and would yield $\kappa' = 1 - \delta_{obs} / \mu_{\delta} = 1 - 10/ 23.333 = .571$ for $v = 1$, and $\kappa' = 1 - \delta_{obs} / \mu_{\delta} = 1 - 150/ 850 = .824$, for $v = 2$. The problem, however, lies in the length of unit 2. Both coders came up with the length of 30 seconds for the second unit, but apparently, the starting and ending points for unit 2 are not the same across coders. For coder 1, the second unit refers to the 30 seconds starting from 1 minute 15 seconds, while for coder 2, the second unit refers to the 30 seconds of content starting from 1 minute. This is an artifact of using length readings directly. In other words, a unit-by-unit comparison of the two coders' length readings can some lead to artificially high agreement values.

One may wonder whether the reformulated κ could provide an easy remedy for this problem, but it turns out that applying the reformulated κ to continuous length data requires a transformation that is even more problematic. The transformation of the length data into a form susceptible to analysis using κ entails the selection of the length or size of a *baseline unit*, or using a familiar discrete content term – an *element*, which has a profound impact on the computation of κ.

Tables 7 and 8 summarize the transformed binary stream for the length readings in Table 6. The baseline unit for Table 7 is 5 seconds, and that for Table 8 is 15 seconds. By applying equations (2) – (8), $\kappa = 1 - \delta_{obs} / \mu_{\delta} = 1 - .174/ .159 = - .095$ for Table 7, and $\kappa = 1 - \delta_{obs} / \mu_{\delta} = 1 - 0.571/ 0.408 = - .400$ for the data in Table 8. Such a large discrepancy clearly demonstrates the fact that applying the reformulated κ to continuous length data is a potentially very bad practice because two researchers using the same data set and the same computational formulae would necessarily arrive at two different values

of κ, unless they choose the same baseline unit that defines the possible breaking points. In fact, if researchers would like to inflate their reliability coefficients, they may simply choose to define 1 second as the baseline unit. The smaller the baseline unit is, the higher κ will be (A simple demonstration is included in Appendix D).

The coefficient based on lengths, on the other hand, is unaffected by the selection of the length of units. The values in the parentheses in Table 6 are expressed in terms of seconds. If the baseline unit is changed to 5 seconds, the value of κ' will still be the same as before: $\kappa' = 1 - \delta_{obs} / \mu_{\delta} = 1 - 2/ 4.667 = .571$ for $v = 1$, and $\kappa' = 1 - \delta_{obs} / \mu_{\delta} = 1 - 6/ 34 = .824$, for $v = 2$. In general, the value of κ' will remain invariant under linear transformations of the dataset. This invariance property is true for all length-based measures. Given the problem with reformulated κ, it seems that a length-based measure that seeks a "moment-by-moment" comparison (Conger, 1985) rather than a unit-by-unit comparison of the lengths should be better that the existing methods. And such is indeed true for the method to be presented in the next section as a remedy for the limited applicability of κ and the inability of κ' to capture hidden disagreement such as unit 2 in Table 6.

A Measure Based on Cumulative Lengths

By borrowing the concept of empirical cumulative distribution function (ECDF) from elementary mathematical statistics, one can envision the disagreement between two coders when unitizing continuous content as the difference between two cumulative length functions. A plot should help illustrating this point. Figure 1 plots two cumulative length functions for the dataset in Table 5. In Figure 1, the dotted line corresponds to coder 1 and the solid line corresponds to coder 2. When coder 2 has used up all available

content, the cumulative length is 70.0 and it remains at 70.0 regardless of how many missing cells there may be. Thus the conceptual problem of having to impute zero(s) for one or more missing cells in the last a couple of columns due to unequal number of units is no longer a concern here, because the cumulative length function is a step function that has its maximum equal to the total length of the content segment. In a sense, for a coder who ends up having more units than the other coders, the representation using cumulative lengths is straightforward, i.e., all the other step functions max-out earlier than the one with more units. Using cumulative lengths also solves the problem of κ' directly. In Table 6, even though the two coders agreed on the length of unit 2, the cumulative lengths are still off by 15 seconds, reflecting the fact that they came up with different starting points for unit 2. In addition, the simulation study to be reported in the next part of this paper will demonstrate that this measure based on cumulative lengths provides some improvement over κ and κ' in terms of correcting for the underestimate of agreement in co-termination.

Using similar notations, let x_{ip} represent the value from the cell corresponding to the ith coder and pth unit, where $i = 1, \ldots, b$, and $p = 1, \ldots, g_{max}$, where g_{max} is the maximum number of units given by the coders. One may consider $g_{max} = \max(g_1, g_2, \ldots, g_b)$, and the squared distance ($v = 2$) between two cumulative lengths corresponding to cells x_{ip} and x_{jp} in a table like Table 5 or Table 6 can be expressed as

$$\Delta^{c}_{ip,jp} = \left[\left(\sum_{k=1}^{p} x_{ik}\right) - \left(\sum_{k=1}^{p} x_{jk}\right)\right]^2 = \left[\sum_{k=1}^{p} (x_{ik} - x_{jk})\right]^2, \qquad (17)$$

and the overall observed disagreement is

$$\delta_{obs}^{c} = \binom{b}{2}^{-1} \sum_{i<j} \sum_{p=1}^{g_{max}} \Delta_{ip,jp}^{c} , \qquad (18)$$

where $i < j$ denotes the sum over all i and j such that $1 \leq i < j \leq b$.

To find the expected disagreement, three steps are involved: 1) conduct a restricted permutation of the unit length data within each block, still holding the missing cells due to the unequal number of units as fixed; 2) for each permutation, calculate cumulative lengths for each cell, 3) use equation (17) and (18) to find the disagreement for that particular permutation, δ^{c}. Here the total number of permutations – M – is the same as given in equation (9). One can imagine conducting an exhaustive permutation to iterate through the M possibilities, and then the expected disagreement is found by dividing the sum of all the δ^{c}'s by M, just as in equation (5).

The squared distance is used in equation (17), simply because when the differences are squared, the complex distance function between two cumulative lengths can be written as a linear combination of the squared distances between simple unit lengths and some cross-product terms. The end result is a solution that provides the exact expected disagreement, i.e. the exact first moment of the permutation distribution of δ_{c}'s, without having to know the value of every element in the distribution. Such a vast reduction in computation is not available for metric Euclidean distances. As a diversion from the main theme, it is worth noting here that for a permutation-based statistic, one can either generate 1) an exact reference distribution, 2) an approximate reference distribution based on random sampling from all possible permutations, or 3) a moment approximation of the reference distribution using the exact values of lower order moments. The first approach is generally computationally infeasible under most

circumstances, and the use of the second approach also requires a large number of repeated random samples in order to achieve stability in the result, otherwise, researchers using the same dataset would necessarily produce different results due to sampling variability. When working under the traditional hypothesis-testing framework, the third approach is often more efficient than the first two approaches, and it is the basis of Berry et al.'s (1976) Pearson Type III distribution approximation to the null distribution of the MRPP test statistic, from which the reformulated κ, κ' and this new measure based on cumulative lengths are derived. Computational formulae for finding the expected disagreement without actually having to conduct permutations are presented as equations (19) – (27). These formulae are still quite cumbersome, but a GAUSS (Aptech Systems Inc., 1997) procedure that implements this method is available in Appendix E.

Let (x_{ip}) and (x_{jq}) denote two rows of unit length data, not cumulative length data, from coders i, and j, where $p, q = 1, \ldots, g_i$, and $1 \le i < j \le b$. Without loss of generality, it is further assumed that g_i is always greater than or equal to g_j. In practice, this restriction is not a concern because the distance function is symmetric and thus one can easily swap the two row vectors to make g_i and g_j satisfy the condition specified. Let $\Delta_{ip,jq}$ – the squared distance between two lengths (cells) – be given as in equation (8), and M be given as in equation (9), define the following identities:

$$D_1(i,j) = \frac{M}{g_i g_j} \sum_{m=1}^{g_j} \left[(g_i + 1 - m) \sum_{q=1}^{g_j} \sum_{p=1}^{g_i} \Delta_{ip,jq} \right]. \tag{19}$$

For $g_j \ge 2$, define:

$$D_2(i,j)=\frac{4M}{g_i g_j(g_i-1)(g_j-1)}\sum_{r=1}^{g_j-1}\sum_{s=r+1}^{g_j}\left[(g_i+1-s)\sum_{q=1}^{g_j-1}\sum_{p=1}^{g_i}\sum_{n>q}^{g_j}\sum_{m\neq p}^{g_i}(\Delta_{ip,jq}\Delta_{im,jn})^{1/2}\right], \quad (20)$$

and for $(g_i - g_j) \geq 1$, define:

$$D_3(i,j)=\frac{M}{g_i}\sum_{m=g_j+1}^{g_i}\left[(g_i+1-m)\sum_{p=1}^{g_i}\Delta_{ip,jm}\right], \quad (21)$$

$$D_4(i,j)=\frac{2M}{g_i g_j(g_i-1)}\sum_{r=1}^{g_j}\sum_{s=g_j+1}^{g_i}\left[(g_i+1-s)\sum_{q=1}^{g_j}\sum_{p=1}^{g_i}\sum_{m\neq p}^{g_i}(\Delta_{ip,jq}\Delta_{im,js})^{1/2}\right], \quad (22)$$

$$D_5(i,j)=\frac{2M}{g_i(g_i-1)}\sum_{r=g_j+1}^{g_i-1}\sum_{s=r+1}^{g_i}\left[(g_i+1-s)\sum_{p=1}^{g_i}\sum_{q\neq p}^{g_i}(\Delta_{ip,jr}\Delta_{iq,js})^{1/2}\right]. \quad (23)$$

The expected disagreement can be readily calculated using the following equations

$$E(i,j)=D_1(i,j)+\psi_1(g_j)D_2(i,j)+\psi_2(g_i,g_j)\{D_3(i,j)+D_4(i,j)+D_5(i,j)\}, \quad (24)$$

where $\psi_1(g_j)$ and $\psi_2(g_i,g_j)$ are two indicator functions of the form:

$$\psi_1(g_j)=\begin{cases}1 & \text{if } g_j \geq 2\\ 0 & \text{if } g_j < 2\end{cases} \quad (25)$$

$$\psi_2(g_i,g_j)=\begin{cases}1 & \text{if } g_i > g_j\\ 0 & \text{if } g_i = g_j\end{cases} \quad (26)$$

and finally

$$\mu_\delta^c=\left[M\binom{b}{2}\right]^{-1}\sum_{i<j}E(i,j). \quad (27)$$

Then the new measure κ^* is given by

$$\kappa^*=1-\delta_{obs}^c/\mu_\delta^c. \quad (28)$$

The numerical results for the example datasets in Tables 4 and 5 are as follows: $\kappa^* = 1 - \delta^c_{obs} / \mu^c_{\delta} = 1 - 1 / 10.222 = .902$, and $\kappa^* = 1 - \delta^c_{obs} / \mu^c_{\delta} = 1 - 143.43 / 551.197 = .74$. For the dataset in Table 6, if 1 second is chosen as the baseline unit, the result is $\kappa^* = 1 - \delta^c_{obs} / \mu^c_{\delta} = 1 - 450 / 1700 = .735$. If 5 seconds is chosen as the baseline unit, the result is $\kappa^* = 1 - \delta^c_{obs} / \mu^c_{\delta} = 1 - 18 / 68 = .735$. If 15 seconds is chosen as the baseline unit, the result is still $\kappa^* = 1 - \delta^c_{obs} / \mu^c_{\delta} = 1 - 2 / 7.556 = .735$. It is easy to see that κ^* is invariant under linear transformations.

Tests of Significance

Because κ is merely a linear function of δ_{obs}, a test of significance of κ is equivalent to the test of δ_{obs}. Mielke and Iyer (1982) gave formulae for the first three moments of the MRBP null distribution, and using the mean and variance, δ_{obs} can be standardized and the associated probability of δ_{obs} can be approximated via a Pearson type III distribution (see Mielke and Berry, 2001). This *p*-value is associated with the test whether κ is significantly different from zero. There is no random sampling assumption involved, and this test of significance is non-asymptotic. Each one of the *n* segments in a reliability study would therefore have a *p*-value, and by looking at the set of *p*-values, the researcher should be able to infer whether the coders' overall agreement is due to chance or not.

A test of significance may be conducted for the coefficient that uses the cumulative lengths via a random sample of all possible permutations (see Edgington, 1987). The exact moments of the null distribution can be derived along the same lines as

equations (19) – (23), but algebra would be very tedious. Generally one is better off leaving the computation to a powerful computer rather than relying on one's analytical skills when the time devoted to solving a particular problem analytically is exponentially greater than what would take for a computer to obtain an answer approximately.

CHAPTER 6

SIMULATION STUDIES

The simulation studies reported here are primarily intended for demonstration purposes. First, it would be ideal to show that the newer methods, κ' and κ^*, especially the latter, which is based on cumulative lengths, are better than the original MRBP reformulated κ in terms of being less conservative. If one of the newer methods can handle discrete data as well as κ or better than κ, and at the same time have the capability to handle continuous length readings, which κ cannot deal with, it should be the method of choice. Second, there are some unknowns about κ' and κ^*'s underlying permutation structure to obtain expected disagreement by permuting lengths rather than by permuting the binary streams of breaking points.

Simulation I: Performance comparisons

The main purpose of this set of simulations was to examine the relative performance of the four competing methods, namely, κ, κ' ($v = 1$), κ' ($v = 2$), and κ^*, under various conditions. The factors manipulated were: 1) number of coders (2, 4, 6), 2) length of the content segments (10, 15, 20, 30), 3) number of units, or equivalently, the number of breaking points plus one (4 to 17, depending on the length of the content), and 4) the proportion of unit boundaries set to be in agreement across coders (0, .5, and .8).

In a sense, this last manipulation creates a "theoretical" value of agreement, and in a sampling experiment, whether a method is conservative or not is determined by comparing the long run average of the agreement coefficients computed using such a method with the theoretical value. Entries in Table 9 to Table 12 are based on the results from a sampling experiment program written in GAUSS (Aptech Systems Inc., 1997). Due to the exponentially increasing amount of computation time, the number of replications for each condition was set to a mere 400, a number usually considered too small, but in this case the performance patterns are quite distinguishable. The program made use of the procedures listed in Appendices B, D, and E. It was assumed that the coders agreed on the number of units contained in any content segment, so what was generally left random was the choice of breaking points in the content streams. Recall that when the numbers of units are equal across coders, co-termination takes a strong form. The data generation process mimics this by creating a series of random binary streams. The number of streams is equal to the number of coders. The length of content is equal to the length of the binary stream plus one. The number of breaking points in the content segment is equal to the number of 1's in the stream. Setting common unit boundaries is a little trickier. For illustrative purposes, in a bivariate (2-coder) case, the two coders are assumed to be co-terminus for a certain number of units, say, n, with n determined by the proportion of units in agreement. If this proportion is zero, no special handling is needed. If the proportion is above zero, n 1's shall be inserted into a random binary stream to create agreement. Note that the data generation process deliberately disadvantaged κ' and κ^* because the content is inherently discrete, which can be handled quite well by κ.

The first line in each one of the cells in Table 9 corresponds to the mean values of the four competing intercoder agreement indices when there are 2 coders and the unit breaking points are completely random, i.e. the proportion of common breaking points is set to zero. In general, regardless of how many breaking points there are or how long the content segment is, all methods yielded long run average coefficients close to zero, which is exactly what is expected of an unbiased method of computing agreement. The standard deviations of the empirical distributions of the four coefficients are also reported for the conditions in Tables 9 to 11. Note that the variability of κ^* is generally larger than that of κ, which is not a surprise because 1) κ^* is defined in a squared Euclidean space, whereas κ utilizes metric Euclidean distance, while this difference is irrelevant to computation of the mean values, it does affect the variance of the distributions; 2) previous literature on the null distributions of MRPP family of statistics (Mielke, 1979) as well as simulation results not reported here indicates that the distribution of κ^* is almost invariably skewed to the negative direction, and a few outlying observations in a skewed distribution will certainly inflate the variance to a great extent.

In Table 10, the proportion of common breaking points is set to 50%. Here the difference between the four methods starts to emerge. Except for those conditions in which the number of breaking points is small (e.g., 4), the only method that consistently kept the agreement coefficient close to the theoretical value (.5) across the board in almost every condition is κ^*.

Table 11 is constructed in much the same way as Tables 9 and 10, only that the theoretical value of agreement is set to .8. The pattern observed in Tables 9 and 10 carries on to Table 11, and the clear winner is again κ^*. The more interesting finding in

Table 11 is that the theoretical agreement is set to a value considered by many (e.g. Krippendorff, 1980) as a rule of thumb to judge whether the coding process is reliable or not. This kind of decision rule has very strong impact on the final outcome of any particular content analysis. Oftentimes, after a researcher has determined that the intercoder agreement coefficient is below .8 (or using some other similar standards), he or she would go back to re-train the coders or use every means possible to improve the coding scheme, and hope that in the next round of reliability assessment, the agreement index will go above .8. Under the current context, if a researcher applied κ to compute an intercoder agreement index for the unitizing phase, he or she is likely to find that it is well below what is commonly expected. Indeed, for the data in Table 11, 80% of all breaking points are set to be in agreement for the two coders, and yet in none of the conditions the value of κ is close to .8.

For four coders, κ^* is still better than κ, but the advantage of κ^* over κ is not as great as seen for the two-coder case. In fact, in some cases κ^* is even worse than κ. A similar trend exists in the six-coder case. The findings are summarized in Tables 12 and 13.

What can be inferred from the simulations reported here is that for two coders, κ is indeed biased downward, and κ^* is clearly the method of choice, but κ starts to pick up when the number of coders increases. These findings are for discrete type of content only, and the combinations of lengths and number of breaking points examined here are perhaps not very representative of what researchers may encounter in the "real world," so making generalizations is quite difficult. However, κ^* does possess the potential to best capture the disagreement of co-termination and its use is encouraged.

Simulation II: What to permute?

In determining the expected disagreement for all length-based measures, including κ' and κ^*, the within-block permutations are conducted using the length data. If it is fair to say that the permutation of the unit lengths is almost inevitable for truly continuous length-readings, such as the data in Table 6, the permutation of lengths transformed from binary streams is quite questionable. Take the unit lengths in Table 4 as an example. After all, the lengths contain the same information as the binary streams in Table 1, so why not conduct the permutations using the binary streams instead of the lengths? Conceptually it is very simple: each permutation of the underlying binary streams shall be re-expressed using lengths, and the length difference or the cumulative length difference shall be taken and recorded to form a permutation distribution from which the expected difference can be calculated.

The answer to this question is not very straightforward. It is certainly possible to conduct permutation on the underlying binary streams, but there are several limitations to this approach. First, expressions of the exact expected disagreement for length-based measures are no longer available (at least for now) if what is permuted is the underlying binary stream. The derivation of the exact first moment of the permutation distributions of MRBP based statistics primarily utilizes the fact that the distances are calculated from the original data that are being permuted. Any transformation of the original data will result in extremely complex solutions. For example, when κ^* is defined as a function of the discrepancies between the cumulative lengths the expression of its expected disagreement is extremely complex, but it is still possible to write out those equations because the cumulative lengths is only a very simple linear combination of the unit

lengths. With the transformation of binary streams to unit lengths, it is already quite impossible to describe the exact nature of such a transformation mathematically, not to mention deriving a solution to find the expected disagreement. Second, simulation data reported in Table 14 show that the long run average of expected disagreement using length permutations will converge to the value obtained via permuting the underlying binary streams.

In Table 14, the long run expected disagreement of the two versions of κ', which are based on permuting lengths rather than the underlying binary streams are reported. The data generation process is the same as for Simulation I, and the number of replications was set to 1,000. The Model I columns are the expected disagreement obtained via permuting the binary streams and they are invariant over the 1,000 replications, so in other words, given the length and the number of breaking points, the expected disagreement is completely determined for Model I permutations. Model II refers to the permutations of lengths, and it is conceivable that for any particular trial in those 1,000 trials, the permutation distribution is a subset of Model I permutation distribution. Overall, the two converge, as can be seen in Table 14: the Model I and Model II column values are very close.

CHAPTER 7

DISCUSSION

The assessment of *co-termination* in the unitizing phase of content analysis is an important issue for communication researchers. The agreement of unitizing focuses on how independent coders choose breaking points at various places in a *segment* of textual content. It is assumed that segments are of two types, discrete and continuous. Discrete segments are composed of a finite number of *elements*, and unitizing a discrete segment is an operation of grouping elements into units, with the *unit* defined as a subset of a segment. For continuous content, it is hard to define what an element is, but it is easy to deal with the *length* of units, most notably expressed in terms of time.

Coefficients of co-termination under various circumstances were considered in this thesis. Generally speaking, for genuinely discrete data, the MRBP reformulated κ can be used directly. The number of coding decisions in this case is the number of elements in the discrete segment minus one, and the reliability data would be binary streams of 0's and 1's. This approach, however, often results in underestimates of agreement, as shown by the simulation studies. Furthermore, this approach cannot be easily extended to deal with continuous content segments. Applying κ to continuous content entails a transformation of the length readings using some pre-defined baseline

unit, and because the value of κ would change depending on the size of the baseline unit, κ is not invariant under linear transformations of the length dataset. The coefficients that are based on lengths, on the other end, can handle the continuous type of content quite well. It is perhaps better to state that they were inspired by the MRBP layout rather than MRBP itself, and with some modification, the new coefficient κ' can handle the situation of coders disagreeing on the number of units contained in the segment, but the inherent problem is that it cannot detect a form of hidden disagreement as exemplified in Table 6, due to that fact that it is comparing length readings in a unit-by-unit fashion. The nice property associated with the length-based measures is that they are invariant under linear transformations of the data, which is absent in MRBP reformulated κ. The last coefficient – κ^*, that makes use of cumulative lengths is not only less conservative than most other indices, it can also solve the inherent problems associated with κ and κ' because the comparison of cumulative lengths is a moment-by-moment comparison of the codings from pairs of coders. Simulation studies were carried out and it was found that κ^* was indeed closer to the theoretical value of agreement than other indices in most situations, especially with 2 coders. The second simulation also justified the permutation structure of the methods based on the permutation of lengths rather than the permutation of the underlying binary streams. Depending on the nature of the study, researchers now possess a family of intercoder agreement indices for the unitizing phase of content analysis, based on the Multi-response Randomized Blocks Permutation procedure. One can imagine that a researcher 1) take a sample of segments from the pool of content, 2) obtain the reliability data for unitizing from a group of independent coders, 3) for each segment, calculate the co-termination index by using one of the methods

discussed in this thesis, and 4) finally, the overall co-termination is found by averaging the co-termination indices over all segments.

As an endnote, I would like to point out that all the indices of co-termination discussed thus far share the same guideline, i.e. they all seek some form of chance-based correction of the observed agreement. The formulation of all these methods share the form of $1 - D_o / D_e$, where D_o is the observed disagreement and D_e is the expected disagreement by chance. A conceptually much simpler extension of the κ^* method may take the form of $1 - D_o / D_m$, where D_m is the maximum disagreement between pairs of coders given the lengths data available. In terms of the MRBP permutations, this D_m is maxima of the permutation distribution of cumulative lengths. Graphically, this can be thought of as the maximum discrepancy between the two cumulative length functions. When the observed disagreement is zero, the agreement index is 1, and when the observed disagreement is equal to the maximum disagreement, agreement is 0. In most cases, the observed discrepancy between the two cumulative length functions falls somewhere in between the extremes, and the value of agreement calculated in this way will necessarily be larger than a chance corrected index such as κ^*. At present, however, the properties of this new index still remain to be uncovered in future research.

LIST OF REFERENCES

Aptech Systems, Inc. (1997). GAUSS (version 3.2.30) [computer program]. Maple Valley, WA: Author.

Berelson, B. (1952). *Content analysis in communication research*. Glencoe, IL: Free Press.

Berry, K. J., & Mielke, P. W. (1988). A generalization of Cohen's kappa agreement measure to interval measurement and multiple raters. *Educational and Psychological Measurement, 48*, 921-933.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20*, 37-46.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*, 213-220.

Conger, A. J. (1985). Kappa reliabilities for continuous behaviors and events. *Educational and Psychological Measurement, 45*, 861-868.

Craig, R. T. (1981). Generalization of Scott's index of intercoder agreement. *Public Opinion Quarterly, 45*, 260-264.

Edgington, E. S. (1987). *Randomization tests* (2nd ed.). New York, NY: Marcel Dekker.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin, 76*, 378-382.

Fleiss, J. L., & Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement, 33*, 613-619.

Fleiss, J. L., Nee, J. C. M., & Landis, J. R. (1979). Large sample variance of kappa in the case of different sets of raters. *Psychological Bulletin, 86*, 974-977.

Fowler, G. L. (1986). Content and teacher characteristics for master's level research course. *Journalism Quarterly, 63*, 594-599.

Frey, L. R., Botan, C. H., & Kreps, G. L. (2000). *Investigating communication: An introduction to research methods* (2nd ed.). Boston: Allyn & Bacon.

Goodman, L. A., & Kruskal, W. H. (1954). Measures of association for cross classifications. *Journal of the American Statistical Association, 49*, 732-764.

Guetzkow, H. (1950). Unitizing and categorizing problems in coding qualitative data. *Journal of Clinical Psychology, 6*, 47-58.

Holsti, O. R. (1969). *Content analysis for the social sciences and humanities*. Reading, MA: Addison-Wesley.

Hubert, L. (1977). Kappa revisited. *Psychological Bulletin, 84*, 289-297.

Janson, H., & Olsson, U. (2001). A measure of agreement for interval or nominal multivariate observations. *Educational and Psychological Measurement, 61*, 277-289.

Krippendorff, K. (1970). Bivariate agreement coefficients for reliability of data. *Sociological Methodology, 2*, 139-150.

Krippendorff, K. (1980). *Content analysis: An introduction to its methodology*. Newbury Park, CA: Sage.

Lombard, M., Snyder-Duch, J., & Bracken, C. C. (2002). Content analysis in mass communication: Assessment and reporting of intercoder reliability. *Human Communication Research, 28*, 587-604.

Mielke, P. W. (1979). On asymptotic non-normality of null distributions of MRPP statistics. *Communications in Statistics – Theory and Methods, A8*, 1541-1550.

Mielke, P. W. (1984). Meteorological applications of permutation techniques based on distance functions. In P. R. Krishnaiah and P. K. Sen (Eds.), *Handbook of statistics, volume 4* (pp. 813-830). Amsterdam: North-Holland.

Mielke, P. W., & Berry, K. J. (1994). Permutation tests for common locations among samples with unequal variances. *Journal of Educational and Behavioral Statistics, 19*, 217-236.

Mielke, P. W., & Berry, K. J. (2001). *Permutation methods: A distance function approach*. New York, NY: Springer-Verlag.

Mielke, P. W., & Iyer, H. K. (1982). Permutation techniques for analyzing multi-response data from randomized block experiments. *Communications in Statistics – Theory and Methods, 11*, 1427-1437.

Mielke, P. W., Berry, K. J., & Johnson, E. S. (1976). Multi-response permutation procedures for *a priori* classifications. *Communications in Statistics – Theory and Methods, A5*, 1409-1424.

Moffett, E. A., & Dominick, J. R. (1987). Statistical analysis in the JOB 1970-85: An update. *Feedback, 28*, 13-16.

Potter, W. J., & Levine-Donnerstein, D. (1999). Rethinking validity and reliability in content analysis. *Journal of Applied Communication Research, 27*, 258-284.

Riffe, D., & Freitag, A. A. (1998). A content analysis of content analysis: Twenty-five years of Journalism Quarterly. *Journalism and Mass Communication Quarterly, 74*, 873-882.

Schaefer, T. M. (1999). The "rhetorical presidency" meets the press: The New York Times and the State of the Union message. *Journalism and Mass Communication Quarterly, 76*, 516-530.

Scott, W. A. (1955). Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quarterly, 19*, 321-325.

Stewart, T. D. (2002). *Principles of research in communication*. Boston: Allyn & Bacon.

Wilhoit, F. G. (1984). Student research productivity: Analysis of Journalism Abstracts. *Journalism Quarterly, 61*, 655-661.

Wimmer, R. D., & Dominick, J. R. (1994). *Mass media research: An introduction* (4th ed.). Belmont, CA: Wadsworth.

APPENDIX A

Permutation tests represent the "ideal" situations where one can derive the exact probabilities rather than approximate values obtained from common probability distributions, such as the t, F and χ^2 (Mielke & Berry, 2001, p. 1). Carrying out randomization or permutation of the collected data rather than relying on the often times unreasonable assumption of random sampling or normality not only make a test more data-dependent, but also enhances the practicability of a test, as the practitioners have full control over the stochastic component of the statistical model.

Assuming an r-dimensional k group design with the combined sample size equaling N, and group sizes equaling n_i where $i = 1, \ldots, k$, and $\sum_{i=1}^{k} n_i = N$, let $(x_{1I}, \ldots, x_{rI})$ denote the r-dimensional responses where $I = 1, \ldots, N$, and let S_i, where $i = 1, \ldots, k$ denote the k groups of responses, or using the terms of Mielke and Berry (2001), the "exhaustive partitioning" of the N responses into k disjoint sets (p. 12). The basic formulation of the MRPP family of statistics involves the definition of a symmetric distance function of the form

$$\Delta_{I,J} = \left[\sum_{c=1}^{r} (x_{cI} - x_{cJ})^2 \right]^{v/2},$$

as a measure of the multivariate distance between the two observations x_I and x_J. For notational simplicity both the "excess group" and the truncation of distance to a preset maximum value shall not be discussed in the present paper (for details see Mielke & Berry, 2001). The choice of v is arbitrary, but the two choices $v = 1$ and $v = 2$ seems most reasonable. When $v = 1$, the distance is metric Euclidean distance and this distance function has nice theoretical properties of being robust and much less influenced by outliers (Mielke & Berry, 2001). When $v = 2$, the distance is defined in a non-metric squared Euclidean space because the triangle inequality fails in this analysis space, and it is known through both theoretical and simulative studies that this choice leads to a less robust test (Mielke & Berry, 1994). However, the choice of $v = 2$ yields an easier explanation of the test results, because many popular tests essentially involve the use of squared distance.

The MRPP statistic can be thought of as a weighted average of within-group distances. Intuitively, a smaller value of the MRPP statistic would mean higher concentration within each *a priori* classified group (Mielke, 1984, p. 815). Such an interpretation is also in line with the geometric interpretation of the conventional multivariate analysis of variance (see Edgington, 1987, pp. 190). Therefore, in terms of detecting between group differences, a smaller value of the MRPP statistic is necessarily "better."

The MRPP statistic is given by

$$\delta_{obs} = \sum_{i=1}^{k} C_i \xi_i ,$$

where C_i is the group weight for $i = 1, \ldots, k$, and $\sum_{i=1}^{k} C_i = 1$, and

$$\xi_i = \binom{n_i}{2}^{-1} \sum_{I<J} \Delta_{I,J} \psi(x_I)\, \psi(x_J),$$

is the average within-group distance for all distinct pairs of responses in the *i*th group. $\psi(\cdot)$ is an indicator function given by

$$\psi(x_I) = \begin{cases} 1 & \text{if } x_I \in S_i, \\ 0 & \text{if } x_I \notin S_i. \end{cases}$$

The choice of group weights is discussed in Mielke (1984), but $C_i = n_i / N$, and $C_i = (n_i - 1) / (N - k)$ are two sensible choices for $v = 1$ and $v = 2$, respectively.

The formal test of significance of δ_{obs} is carried out by assuming the null hypothesis of equal probabilities being placed upon each one of the

$$M = \frac{N!}{\prod_{i=1}^{k} n_i!},$$

possible permutations of the N responses into the k groups, each permutation yielding a realized value of δ. The probability value associated with δ_{obs} is a ratio of the number of δ s being smaller than or equal to δ_{obs} and M, formally written as $P(\delta_{obs}) = \{\#\ \delta \le \delta_{obs}\} / M$.

Because M is usually a very large number even for relatively small sample sizes, the exact reference distribution of the MRPP statistic is difficult to obtain, therefore, Mielke, Berry and Johnson (1976) have provided efficient computational methods for the first three cumulants of the MRPP null distribution, upon which a moment approximation using Pearson type III distribution may be utilized. Generally this approximation is excellent. For details please refer to Mielke & Berry (2001).

APPENDIX B

The following is a GAUSS (Aptech Systems Inc., 1997) procedure that implements the multi-coder version of κ. The input is a *b*-coder by *g*-unit matrix of binary streams similar to Table 1. The output is a 3-by-1 vector, call it *resultv*, with the first element being the observed disagreement, the second element being the expected disagreement (un-averaged), and the third element being the denominator of expected disagreement. So agreement is simply 1 – *resultv*[1]/(*resultv*[2]/ *resultv*[3]).

```
proc kappa(x);
  local delta, distance, temp, x1, x2, n1, n2, distance1, factor1, resultm;
  delta = 0;
  distance = 0;
  n1 = cols(x);
  n2 = n1;
  for idxI (1, rows(x)-1, 1);
    for idxJ (idxI+1, rows(x), 1);
      x1 = x[idxI,.]';
      x2 = x[idxJ,.]';
      delta = delta + sumc(abs(x1-x2));
      distance1 = 0;
      for i (1, n1, 1);
        for j (1, n2, 1);
          distance1 = distance1+abs(x1[i] - x2[j]);
        endfor;
      endfor;
      distance = distance + distance1;
    endfor;
  endfor;
  resultm = zeros(3,1);
  resultm[1] = delta/n1;
  resultm[2] = distance/n1;
  resultm[3] = n1;
  retp(resultm);
endp;
```

APPENDIX C

The following is a GAUSS (Aptech Systems Inc., 1997) procedure that implements the method κ'. The input is a *b*-coder by *g*-unit matrix similar to Table 5. The output is a 3-by-1 vector, call it *resultv*, with the first element being the observed disagreement, the second element being the expected disagreement (un-averaged), and the third element being the denominator of expected disagreement. So agreement is simply 1 – *resultv*[1]/(*resultv*[2]/ *resultv*[3]).

```
proc kappaprime(x,v);
  local M, delta, distance, temp, x1, x2, n1, n2, distance1, factor1, factor2,
resultm;
  M = prodc(sumc((x .gt 0)')!);
  delta = 0;
  distance = 0;
  for idxI (1, rows(x)-1, 1);
    for idxJ (idxI+1, rows(x), 1);
      x1 = x[idxI,.]';
      x2 = x[idxJ,.]';
      n1 = sumc(x1 .gt 0);
      n2 = sumc(x2 .gt 0);
      if (n1 lt n2);
        temp = n1;
        n1 = n2;
        n2 = temp;
        temp = x1;
        x1 = x2;
        x2 = temp;
      endif;
      x1 = x1[1:n1];
      x2 = x2[1:n1];
      delta = delta + sumc(abs(x1-x2)^v);
      distance1 = 0;
      for i (1, n1, 1);
        for j (1, n2, 1);
          distance1 = distance1+abs(x1[i] - x2[j])^v;
        endfor;
      endfor;
```

```
      factor1 = M*(1/n1);
      factor2 = M*(1-n2/n1);
      distance = distance + distance1*factor1+sumc(x1^v)*factor2;
    endfor;
  endfor;
  resultm = zeros(3,1);
  resultm[1] = delta/(rows(x)!/(rows(x)-2)!/2);
  resultm[2] = distance/(rows(x)!/(rows(x)-2)!/2);
  resultm[3] = M;
  retp(resultm);
endp;
```

APPENDIX D

First, assuming that there are only 2 coders, the total length of the content segment is N, and the coders agreed on the number of units in the content – denote it by a. In order to transform the continuous content into binary streams to reflect the breaking points and non-breaking points, the length of the baseline unit is first chosen to be n. As a consequence, there are altogether $(N / n - 1)$ elements in the transformed binary streams, and the number of 1's in the two streams are both $(a - 1)$, and the number of 0's in the two streams are both $(N / n - a)$. It is straight forward from equation (7) that the excepted average disagreement is $2 (a - 1) (N / n - a) / n^2$. Denote the observed average disagreement with δ / n, it is again straightforward from equation (2) that $\kappa = 1 - \delta / [2 (a - 1) (N / n - a) / n] = 1 - \delta / \{(2a - 2) [(N - an) / n^2]\}$. It is easy to see that as n decreases, the identity $(N - an) / n^2$ increases, because N and a are constants regardless of how many basic elements there are in the binary stream. The result is therefore the decrease in the length of the baseline unit is directly associated with an increase in κ.

APPENDIX E

The following is a GAUSS (Aptech Systems Inc., 1997) procedure that implements the method based on cumulative lengths, κ^*. The input is a *b*-coder by *g*-unit matrix similar to Tables 5 or 6. The output is a 3-by-1 vector, call it *resultv*, with the first element being the observed disagreement, the second element being the expected disagreement (un-averaged), and the third element being the denominator of expected disagreement. So agreement is simply 1 – *resultv*[1]/(*resultv*[2]/ *resultv*[3]).

```
proc kappastar(x);
  local delta, distance, M, temp, x1, x2, n1, n2, distm, tempdistm, part2,
        indexm, minor, minorv, factor, resultm;
  delta = 0; distance = 0;
  M = prodc(sumc((x .gt 0)')!);
  for idxI (1, rows(x)-1, 1);
    for idxJ (idxI+1, rows(x), 1);
      x1 = x[idxI,.]';
      x2 = x[idxJ,.]';
      n1 = sumc(x1 .gt 0);
      n2 = sumc(x2 .gt 0);
      if (n1 lt n2);
        temp = n1;
        n1 = n2;
        n2 = temp;
        temp = x1;
        x1 = x2;
        x2 = temp;
      endif;
      x1 = x1[1:n1];
      x2 = x2[1:n1];
      for i (1, n1, 1);
        delta = delta + (sumc(x1[1:i])-sumc(x2[1:i]))^2;
      endfor;
      distm = zeros(n1, n1);
      for i (1, n1, 1);
        distm[.,i] = x1[.,1] - x2[i,1];
      endfor;
      for j (1, n2, 1);
        distance = distance+sumc(vec(distm[.,1:n2]^2))*(1/n1/n2)*M*(n1+1-j);
```

```
      endfor;
      if (n2 ge 2);
        for i (1, n2-1, 1);
          for j (i+1, n2, 1);
            tempdistm = distm[.,1:n2];
            for col (1, n2-1, 1);
              for row (1, n1, 1);
                indexm = zeros(n1,n2);
                indexm[row,.] = ones(1,n2);
                minor = delif(tempdistm,indexm);
                minorv = tempdistm[row,col].*vec(minor[.,col+1:n2]);
                factor = 4*(1/n1)*(1/n2)*(((n2-1)*(n1-1))^(-1))*M*(n1+1-j);
                distance = distance + factor*sumc(minorv);
              endfor;
            endfor;
          endfor;
        endfor;
      endif;
      if ((n1 - n2) ge 1);
        for j (n2+1, n1, 1);
          distance = distance + sumc(distm[.,j]^2)*(1/n1)*M*(n1+1-j);
        endfor;
        for i (1, n2, 1);
          for j (n2+1, n1, 1);
            tempdistm = distm[.,1:n2];
            part2 = distm[.,j];
            for col (1, n2, 1);
              for row (1, n1, 1);
                indexm = zeros(n1,1);
                indexm[row] = 1;
                minor = delif(part2,indexm);
                minorv = tempdistm[row, col].*minor;
                factor = 2*(1/n1)*(1/n2)*((n1-1)^(-1))*M*(n1+1-j);
                distance = distance + factor*sumc(minorv);
              endfor;
            endfor;
          endfor;
        endfor;
        for i (n2+1, n1-1, 1);
          for j (i+1, n1, 1);
            tempdistm = distm[.,i];
            part2 = distm[.,j];
            for row (1, n1, 1);
              indexm = zeros(n1,1);
              indexm[row] = 1;
              minor = delif(part2,indexm);
              minorv = tempdistm[row].*minor;
              factor = 2*(1/n1)*((n1-1)^(-1))*M*(n1+1-j);
              distance = distance + factor*sumc(minorv);
            endfor;
          endfor;
        endfor;
      endif;
    endfor;
  endfor;
  resultm = zeros(3,1);
  resultm[1] = delta/(rows(x)!/(rows(x)-2)!/2);
  resultm[2] = distance/(rows(x)!/(rows(x)-2)!/2);
  resultm[3] = M;
  retp(resultm);
endp;
```

APPENDIX F

	Treatments					
Blocks (Coders)	1	2	3	4	5	6
1	1	0	0	0	1	0
2	1	0	0	0	0	1

Table 1
Example dataset: MRBP layout

	Coder 1		
Coder 2	1	0	Sums
1	1/6	1/6	1/3
0	1/6	1/2	2/3
Sums	1/3	2/3	1

Table 2
Cross-classification layout of data in table 1

	Treatments					
Blocks (Coders)	1	2	3	4	5	6
1	0	1	1	0	0	0
2	1	0	0	0	0	1

Table 3
A possible permutation of the data in table 1

	Treatments		
Blocks (Coders)	1	2	3
1	1	4	2
Cumulative 1	1	5	7
2	1	5	1
Cumulative 2	1	6	7

Table 4
Unit lengths: transformation of data in table 1

	Treatments						
Blocks (Coders)	1	2	3	4	5	6	7
1	10.3	19.7	10.1	9.8	4.6	5.0	10.5
Cumulative 1	10.3	30.0	40.1	49.9	54.5	59.5	70.0
2	10.2	19.8	8.5	12.0	9.5	10.0	–
Cumulative 2	10.2	30.0	38.5	50.5	60.0	70.0	70.0

Table 5
Unit lengths: unequal number of units

	Treatments		
Blocks (Coders)	1	2	3
1	1:15 (75)	0:30 (30)	0:15 (15)
Cumulative 1	1:15 (75)	1:45 (105)	2:00 (120)
2	1:00 (60)	0:30 (30)	0:30 (30)
Cumulative 2	1:00 (60)	1:30 (90)	2:00 (120)

Table 6
Unit lengths: hidden disagreement for unit 2

Blocks (Coders)	Treatments																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0

Table 7
Transformation of data in table 6 using 5 seconds as a baseline unit

	Treatments						
Blocks (Coders)	1	2	3	4	5	6	7
1	0	0	0	0	1	0	1
2	0	0	0	1	0	1	0

Table 8
Transformation of data in table 6 using 15 seconds as a baseline unit

		Methods			
Length	Units	κ	κ' ($v = 1$)	κ' ($v = 2$)	κ*
10	4	-.022	.017	.016	-.016
	*SD*s	.342	.352	.528	.781
10	6	-.006	.004	.009	.008
	*SD*s	.362	.316	.404	.746
10	8	-.022	-.036	-.040	-.025
	*SD*s	.346	.330	.361	.737
15	4	.025	.035	.048	.048
	*SD*s	.266	.324	.502	.725
15	10	.010	-.007	-.002	-.053
	*SD*s	.285	.253	.313	.703
20	4	-.004	-.009	-.007	-.011
	*SD*s	.228	.326	.505	.734
20	10	-.023	-.018	-.020	-.004
	*SD*s	.230	.207	.302	.704
20	16	.016	.024	.022	.050
	*SD*s	.241	.226	.249	.759
30	4	-.003	-.002	.001	.002
	*SD*s	.202	.327	.524	.759
30	17	.018	.003	.003	.086
	*SD*s	.190	.161	.224	.819
	Overall Means	-.001	.001	.003	.009

Table 9
Performance comparison for 2 coders: 0 % unit boundaries in agreement

		Methods			
Length	Units	κ	κ' ($v = 1$)	κ' ($v = 2$)	κ*
10	4	.234	.150	.189	.275
	*SD*s	.288	.406	.571	.696
10	6	.204	.147	.179	.344
	*SD*s	.310	.365	.468	.601
10	8	.127	.103	.119	.273
	*SD*s	.373	.366	.401	.710
15	4	.338	.206	.245	.411
	*SD*s	.210	.371	.553	.672
15	10	.172	.142	.165	.405
	*SD*s	.257	.281	.355	.581
20	4	.359	.248	.307	.436
	*SD*s	.173	.369	.559	.692
20	10	.276	.187	.211	.476
	*SD*s	.192	.237	.326	.411
20	16	.137	.118	.132	.359
	*SD*s	.253	.235	.285	.688
30	4	.322	.199	.234	.400
	*SD*s	.130	.367	.539	.620
30	17	.245	.182	.204	.498
	*SD*s	.158	.176	.247	.378
	Overall Means	.241	.168	.200	.388

Table 10
Performance comparison for 2 coders: 50 % unit boundaries in agreement

		Methods			
Length	Units	κ	κ' ($v = 1$)	κ' ($v = 2$)	κ^*
10	4	.573	.419	.510	.667
	*SD*s	.190	.457	.548	.430
10	6	.640	.484	.568	.753
	*SD*s	.184	.400	.453	.330
10	8	.361	.293	.330	.504
	*SD*s	.367	.424	.456	.666
15	4	.579	.393	.471	.691
	*SD*s	.118	.430	.517	.362
15	10	.438	.327	.387	.716
	*SD*s	.179	.303	.366	.281
20	4	.600	.398	.490	.701
	*SD*s	.089	.432	.520	.348
20	10	.590	.424	.502	.793
	*SD*s	.105	.286	.351	.184
20	16	.375	.289	.327	.678
	*SD*s	.217	.281	.344	.381
30	4	.632	.397	.476	.730
	*SD*s	.071	.421	.507	.324
30	17	.516	.368	.427	.802
	*SD*s	.110	.219	.281	.190
	Overall Means	.530	.379	.449	.703

Table 11
Performance comparison for 2 coders: 80 % unit boundaries in agreement

	Methods			
Length	κ	κ' ($v = 1$)	κ' ($v = 2$)	κ^*
0 % unit boundaries in agreement				
10	.000	.003	.004	-.002
15	.004	.006	.004	.014
20	-.006	-.002	-.005	-.013
30	-.002	-.013	-.018	-.017
Total	-.001	-.002	-.003	-.005
50 % unit boundaries in agreement				
10	.210	.101	.115	.178
15	.253	.120	.140	.277
20	.250	.124	.145	.285
30	.308	.146	.169	.328
Total	.255	.123	.142	.254
80 % unit boundaries in agreement				
10	.527	.352	.398	.475
15	.586	.359	.423	.620
20	.581	.363	.425	.651
30	.608	.357	.423	.696
Total	.576	.358	.417	.610

Table 12
Performance comparison for 4 coders

	Methods			
Length	κ	κ' ($v = 1$)	κ' ($v = 2$)	κ^*
0 % unit boundaries in agreement				
10	.000	.002	.001	.003
15	-.001	.001	.004	.013
20	.000	-.002	-.002	.017
30	-.001	-.001	-.002	-.014
Total	-.001	.000	.000	.005
50 % unit boundaries in agreement				
10	.210	.102	.118	.139
15	.256	.116	.134	.214
20	.246	.128	.147	.269
30	.310	.155	.179	.357
Total	.256	.125	.145	.245
80 % unit boundaries in agreement				
10	.526	.350	.394	.459
15	.583	.349	.408	.610
20	.579	.353	.412	.649
30	.607	.352	.413	.691
Total	.574	.351	.407	.602

Table 13
Performance comparison for 6 coders

		κ' ($v = 1$)		κ' ($v = 2$)	
Length	Units	Model I	Model II	Model I	Model II
3	2	0.500	0.500	0.500	0.500
4	2	0.879	0.889	1.312	1.333
5	2	1.237	1.249	2.512	2.496
5	4	0.375	0.375	0.375	0.375
6	2	1.581	1.599	3.950	3.994
6	4	0.660	0.660	0.908	0.899
7	2	1.936	1.945	5.832	5.833
7	4	0.905	0.915	1.533	1.575
7	6	0.278	0.278	0.278	0.278
8	2	2.273	2.286	7.848	8.002
8	4	1.160	1.156	2.418	2.400
8	6	0.497	0.499	0.629	0.635
10	2	2.948	2.956	13.271	13.288
10	4	1.632	1.616	4.561	4.495
10	6	0.881	0.878	1.594	1.588
10	8	0.399	0.400	0.484	0.486
15	2	4.600	4.637	31.846	32.421
15	7	1.384	1.385	3.659	3.679
15	12	0.405	0.406	0.520	0.529
30	2	9.597	9.642	139.497	139.681
30	12	1.792	1.797	6.272	6.343
30	22	0.565	0.565	0.909	0.905
50	2	16.640	16.337	408.768	400.369
50	12	3.448	3.446	22.378	22.336
50	22	1.592	1.597	5.270	5.282
50	32	0.820	0.817	1.675	1.651
50	42	0.327	0.326	0.434	0.432

Table 14
Comparison of expected disagreement

APPENDIX G

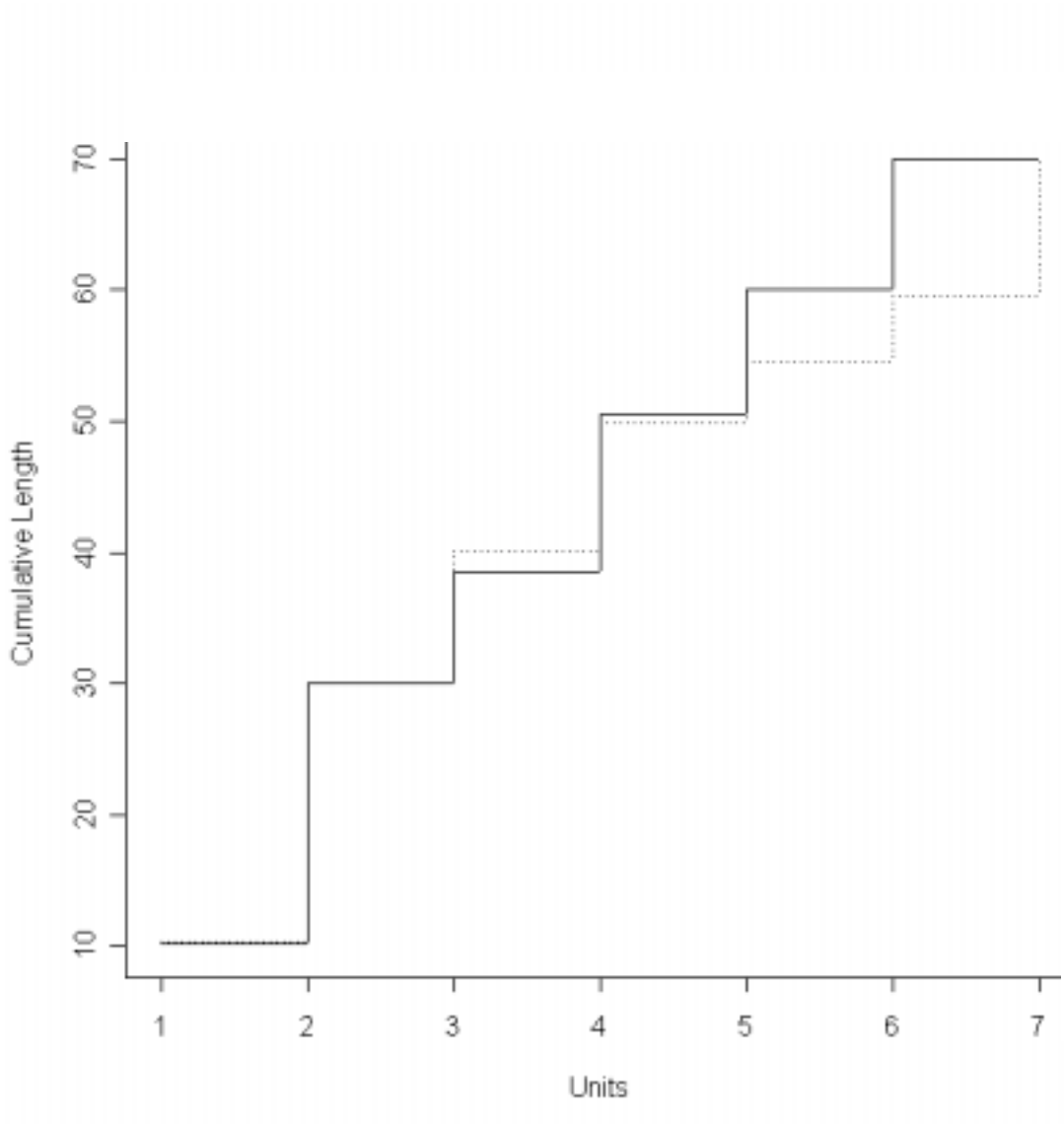

Figure 1
Illustration of two cumulative length functions for data in table 5