PACIFIC LINGUISTICS

Series B - No. 75

EXPERIMENTAL GLOTTOCHRONOLOGY: BASIC METHODS AND RESULTS

by

J.B.M. Guy

Department of Linguistics
Research School of Pacific Studies
THE AUSTRALIAN NATIONAL UNIVERSITY

Guy, J.B.M. *Experimental glottochronology: Basic methods and results.*
B-75, viii + 224 pages. Pacific Linguistics, The Australian National University, 1980. DOI:*10.15144/PL-B75.cover*

PACIFIC LINGUISTICS is issued through the *Linguistic Circle of Canberra* and consists of four series:

SERIES A - OCCASIONAL PAPERS
SERIES B - MONOGRAPHS
SERIES C - BOOKS
SERIES D - SPECIAL PUBLICATIONS

ALL CORRESPONDENCE concerning *PACIFIC LINGUISTICS*, including orders and subscriptions, should be addressed to:

The Secretary,
PACIFIC LINGUISTICS,
Department of Linguistics,
School of Pacific Studies,
The Australian National University,

Canberra, A.C.T. 2600.
Australia.

First published 1980.

The editors are indebted to the Australian National University for help in the production of this series.

This publication was made possible by an initial grant from the Hunter Douglas Fund.

National Library of Australia Card Number and ISBN 0 85883 220 8

This monograph was produced on DEC-KL10 using VIDED, a display-oriented text editor written by Jacob Palme, of the Swedish National Defense Research Institute, and printed out on a Diablo-1640. All software referred to in this monograph was written by the author in SIMULA. Unless otherwise acknowledged, the algorithms presented were developed by the author and are believed to be original. Numbers in square brackets refer to figures and tables in appendix 1.

TABLE OF CONTENTS

Guy, J.B.M. *Experimental glottochronology: Basic methods and results.*
B-75, viii + 224 pages. Pacific Linguistics, The Australian National University, 1980. DOI:*10.15144/PL-B75.cover*

CHAPTER 1: INTRODUCTION

ORIGINS

THE FAILURE OF TRADITIONAL LEXICOSTATISTIC METHODS

The studies in this monograph stem from a feeling of dissatisfaction with classic lexicostatistical methods. I had in 1974 written some software to produce the cognate percentage tables published in Tryon's internal classification of the New Hebrides languages (Tryon 1976:95-162) and the 178x178 table, which would have taken about 30 years to produce by hand, had only taken 30 seconds of the computer's time. And so it was all the more disheartening to discover that we could find no hard and fast set of rules for interpreting it. Much had been written on the subject in linguistic publications, retention rates calculated for long-recorded languages, various methods proposed to overcome the problem posed by fluctuating retention rates, but the validity of glottochronology still remained a matter of faith.

In 1978 Tryon presented a paper for the Second International Conference on Austronesian Linguistics, where he attempted an external classification of the New Hebrides languages, partially based on cognate counts. These, to quote him, produced a "rather undifferentiated picture". This was a kind understatement: where the picture was differentiated enough for a probably significant pattern to emerge, the pattern was wrong (see [1]). Aware of the fact, Tryon presented an alternative chart [2], based on the same method as his internal grouping (Tryon 1976:80, 1978:880), but using lower thresholds, and asked me if I knew of other methods, at which request I produced a minimal-spanning tree [3], using as a measure of the distance between languages linear-correlation coefficients calculated from the computer-produced table of cognate percentages given in his paper.

> The idea of using linear-correlation coefficients as a measure of the genetic affinity between languages had sprung from a question put to me by Tryon in 1976: if two languages show a given percentage of shared cognates, but one of them is represented by a defective or shorter wordlist, how can one estimate the percentage of cognates they would have shown had a full standard wordlist been available? I attempted to find a solution to the problem by selecting from his 178 wordlists pairs of close dialects, one of which was represented by a defective wordlist. For each such pair a scatter diagram was drawn, on which each list not belonging to the pair was plotted according to its

Guy, J.B.M. *Experimental glottochronology: Basic methods and results.*
B-75, viii + 224 pages. Pacific Linguistics, The Australian National University, 1980. DOI:*10.15144/PL-B75.1*

cognate percentages with the members of the pair. For instance, Lembinwen and Benour, two very close dialects, were represented by lists of different sizes. A scatter diagram was plotted, showing only those lists which had 180 items or less in common with the Lembinwen list, and 230 or more with the Benour list. Toga, for instance, with 31.8% cognates with Lembinwen out of a total of 179 common items, and 30.4% cognates with Benour out of a total of 230 items, would be represented by a point about 30.4mm from the Y-coordinates (Benour) and 31.8mm from the X-coordinates (Lembinwen). The resulting graphs showed a very strong linear correlation [4].

The minimum-spanning tree drawn from the correlation coefficients computed on the table of cognate percentages, far from contradicting our intimate understanding of the external relationships of the New Hebrides languages, as did the one obtained from traditional lexicostatistical methods, even showed Xaraci (New Caledonia) and Nengone (Loyalty Group) as forming a close group on their own.

Challenged by Dyen to justify the method used, and unable to do so on theoretical grounds as I had not worked out its implications, I was reduced to saying that the proof of the pudding was in the eating, and that since that method produced results more satisfying to our intuition than any of the others tried so far, it had to be better. An obvious answer to such an argument is that our intuition must have led us to very wrong conclusions indeed, that they should be so far removed from those of the accepted method. How does one prove a theory to be better than another?

THE NEED FOR CONTROLLED EXPERIMENTS

Glottochronology suffers from much theorizing and little experimenting, and it might at this stage be useful to repeat some principles of the scientific method.

A theory (or formula) predicts the outcome of an event. How closely the prediction fits the actual outcome is a measure of the validity of the theory. A theory is vacuous if no test can be devised that could invalidate it.

Thus the formula for the distance travelled by a mass falling in a vacuum is strictly valid only if the acceleration is constant, which can hardly be true in any practical experiment, even in a room where a perfect vacuum has been obtained, since the gravitational pull of the Earth increases as the mass nears the ground.

A theory, then, applies validly only to a certain range of events (linear travel in a vacuum under constant acceleration), with good enough approximation to a wider range (free fall in a vacuum, fall of small heavy objects in a light atmosphere), and from poorly to not at all beyond (fall of a feather on a windy day).

A physical formula or a theory is elaborated through observation, experimentation, and a good deal of serendipity. Its validity should be tested through experimentation. The experimenter observes or causes an event (i.e. conducts an experiment) and, having predicted its

outcome according to the theory, checks his prediction against the actual outcome. By repeating his observations and by varying the conditions under which they are carried out he ascertains the range of events for which the formula or theory holds.

Most importantly, experiments must be repeatable.

But predictions need not be perfectly accurate (meteorology), and may even be about the causes of events rather than their outcomes (forensic medicine). Glottochronology (the theory) predicts the history of the derivation from a common ancestor (the cause) of a group of interrelated languages reduced to sample wordlists (the event). To test its validity one should conduct repeated experiments, each consisting of three steps:

a) gather sample wordlists from a language family the phylogeny of which is precisely known from historical records,

b) predict its phylogeny using some lexicostatistical method,

c) compare the predicted phylogeny to the actual phylogeny as attested by the records.

THE INADEQUACY OF DOCUMENTED LEXICAL DATA

Language groups for which we have satisfactory historical records are exceedingly few, and therefore not only are experiments not repeatable at will under varied conditions (in which case the linguist would only be in a position not much worse than that of the astronomer observing distant phenomena over which he has no control), but moreover not repeatable for several centuries (the time for the languages recorded to evolve appreciably).

Worse still, the written form of a language is likely to influence the evolution of its spoken form to such an extent that a lexicostatistic method shown to be valid for long-recorded language groups fails to apply to a family of purely spoken languages. There is then no assurance that a lexicostatistic method the validity of which has been amply demonstrated for written languages is also valid for unwritten ones.

The linguist is in an unenviable position.

INVESTIGATING LANGUAGE CHANGE THROUGH COMPUTER SIMULATIONS

ELABORATING A MODEL

The simulation of the lexical diversification of language families appears to be a formidable task, involving so many variables that only

a crude model could at best be devised (cf. Fodor 1965). It had nevertheless been a long-standing pet project of mine which its apparent difficulty and the uncertainty of ever bringing it to a successful completion had given it a very low priority. One of the later models, partially translated into a computer program, took into account a large number of factors.

> A map was randomly created, consisting of randomly connected vertices. To each edge connecting two vertices X and Y were associated two numbers within the range 0 to 1, representing the difficulty of travelling from X to Y and from Y to X. A community was then created, defined by a random number representing its population, and a list of random numbers representing a sample of its communalect. As the community grew in numbers, it became likely to split into two or more communities, some, all, or none of which might then migrate to another location. The probability of a community migrating, and the location to which it would migrate depended on the difficulty of travelling there, on population densities, and on how friendly or inimical its relationships were with other communities at their current location and at their prospective place of migration. Friendliness or enmity between any two communities was expressed by a number within the range -1 to +1, periodically recomputed on the basis of the amount of recent past contacts, and available resources. Vocabulary retention rates and word-borrowing were again a complex function of friendliness/enmity, geographical proximity, stress (itself a function of population densities and the state of relations between neighbouring communities), and divergence between communalects.

It then dawned upon me that the model I was painstakingly developing and implementing completely missed the point.

Glottochronology aims at reconstructing the history of the lexical innovations of languages, not at discovering the causes of those changes. Everything in the model relating to the causes of those changes (geographical location, population densities, state of relations between communities, etc.) was irrelevant to the problem. As soon as this had become clear, it took just a week to devise a simple language in which to instruct a computer to create and simulate the diversification of a list of items symbolizing a standard wordlist of the type used in glottochronology.

IMPLEMENTATION

The simulation language designed for the purpose was exceedingly simple, consisting of just four reserved words (CREATE, TIME, SPLIT, REPORT). Wordlists were not allowed to borrow from one another, and their retention rate was kept constant in time and equal for all items, as it was reasoned that the performance of current lexicostatistical methods should first be judged on the basis of the

assumptions underlying them: that vocabulary is replaced at a constant rate, and that loanwords are weeded out. Later, further commands were to be added to the simulation language to investigate the effects of borrowing and varying retention rates.

That the retention rate is kept constant does not mean that after 1000 years with a retention rate of 50% per 1000yrs a 100-word lexicon will have retained exactly 50 of its original items (such an event is as unlikely as a hundred tosses of a fair coin resulting in exactly 50 heads and 50 tails), and the computer program to interpret the simulation language was of course designed accordingly.

> Wordlists are represented by arrays of real numbers the integer part of which represents a gloss, and the decimal part of which expresses the retention rate of that item. Whenever a wordlist is to be updated (i.e. on encountering the commands SPLIT and REPORT), the time elapsed since it was last updated is computed; the list is then examined item by item, and the retention rate of each item brought to the power of the time elapsed, giving the probability of the item being retained over that time span; a random number from 0 to 1 is then generated; if it happens to be greater than the probability of retention just computed, the integer part of the real number representing that item in that list is replaced by a new, randomly-generated integer.

The simulation method is best shown through an example. Here is, with an interlinear plain English translation, a short sequence of commands to create a family of languages:

```
CREATE: AZ SPLIT: AZ ALPHA BG HJ
Create a language AZ, and let AZ split into ALPHA, BG, and HJ

TIME: 500 SPLIT: BG BD EG
in 500 let BG split into BD and EG

TIME: 1000 SPLIT: EG EF GOLF
in 1000 let EG split into EF and GOLF

TIME: 1100 SPLIT: BD BRAVO CD
in 1100 let BD split into BRAVO and CD

TIME: 1500 SPLIT: CD CHARLIE DELTA
in 1500 let CD split into CHARLIE and DELTA

SPLIT: HJ HOTEL INDIA JULIET
and HJ split into HOTEL, INDIA, and JULIET

TIME: 1700 SPLIT: EF ECHO FOXTROT
in 1700 let EF split into ECHO and FOXTROT

TIME: 1980 REPORT: L1980
in 1980 make a copy of all extant wordlists (i.e. ALPHA through
to JULIET) and file it under the name L1980.
```

When the simulation program is run the following exchange takes place first between the computer and the user (the computer's contribution to the dialogue is underlined):

Size of wordlists: 100
Retention rate: 0.9
Accuracy: 5000
Innovations to be listed into file: INNOV
Instructions are in file: TEST

The standard wordlist has here been defined as containing 100 items, and the retention rate set to 90% per 1000yrs. The accuracy with which cognates are recognized is simulated by allowing only a given number (here 5000) of possible different word shapes, for whereas one can assert beyond reasonable doubt 'these are not cognates' it is rarely possible to assert 'those are'; thus for instance Proto-North New Hebridean iga 'fish' would regularly yield e in Lehali; that we do find Lehali e meaning 'fish' does not, however, prove beyond reasonable doubt that it derived from iga since it might just as regularly have derived from ige, ego, ege, ego, ie, etc.... By limiting the number of different possible word shapes, we allow misleading evidence as occurs in reality.

The program then goes on to read and execute the instructions contained in the file named TEST, keeping in a file called INNOV a running account of the evolution of the language family it is creating.

PRELIMINARY EXPERIMENT

A reasonably complex language family of 19 languages ([5], [6]) was created in a preliminary experiment. Wordlist size was set to 40 items, the retention rate to 80% per thousand years, and the accuracy of cognate recognition to 1000.

> There was a practical reason for the choice of such a short wordlist: the speed of the programs written to test certain methods was so difficult to evaluate in the abstract that the shortest possible meaningful wordlist had to be specified, lest a large amount of computer time should be unnecessarily wasted.

The phylogenetic tree ([7]) of the language family created by the simulation was then drawn from the running account of its evolution kept by the program ([8]). As expected, many deviations from the set rate of vocabulary retention had taken place, some of them quite large, which strengthened the feeling that traditional methods applied here would not lead to a very accurate reconstruction.

From the wordlists (lists of integer numbers) created by the simulation, another computer program produced a table of percentages of shared cognates ([9]). Note that phonological change is not being simulated and that forms are therefore cognate when represented by the same integer.

CHAPTER 2: TRADITIONAL LEXICOSTATISTICAL METHODS

TRANSLATING LEXICOSTATISTICAL METHODS INTO A COMPUTER PROGRAM

A computer program was written to implement those methods to which linguists seemed most likely to resort. The clearest description of one of those methods is found in Ross (1979):

> A tree diagram roughly reflecting diachronic reality is derived [...] by drawing nodes progressively at the highest MINIMUM percentage of cognates shared between two languages, one in each group (Sanders 1977). The first step treats each individual language as a group, and connects it to the language with which it has the highest percentage of shared cognates. Some languages may be omitted from this stage, as their highest shared cognate percentage is with a language, A, which has a higher shared cognate percentage with language B, with which it is therefore to be linked [later]. At the second step, each group (whether of one, two, or more languages) resulting from the first step is linked to the group with which one of its languages has the highest minimum percentage of shared cognates. Thus if the minimum shared cognate percentage of a language in group X with a language in group Y is 35, and the minimum of a language in group X with a language in group Z is 33, then group X and Y are connected at the second step. The third and further steps repeat the same process until all languages are included in the tree.

The method described by Ross, however, is not the only one in common use. Thus, replacing every occurrence of the word 'minimum' by 'maximum' in the above description yields another method, perhaps even more popular, and which appears to be the one followed by Tryon in his external classification of the New Hebrides languages (1978:892ff). A third method, seldom used because it drastically increases the number of calculations to be carried out, is obtained by rewording thus the instructions for the second step in Ross's description:

> At the second step, each group ... resulting from the first step is linked to the group with which it has the highest MEAN percentage of shared cognates. Thus, if the two languages A and B of a group X share 30%, 40%, 42%, and 36% cognates with the two languages C and D of a group Y, and 36% and 44% with the only language of a group Z, then the mean percentage of shared cognates being 37% between group X and group Y (i.e. (30+40+42+36)/4), and 40% between group X and group Z (i.e. (36+40)/2), groups X and Z are connected.

The computer program allowed the user to choose any of the three

methods just described. But the options open to a prospective user of traditional lexicostatistic techniques do not stop here: what of those cases where some language X is found to share almost as many cognates with Y as with Z? Should X, Y, and Z be considered to have split at the same time? For it seems unreasonable indeed to hold that the fact that X shares 50% cognates with Y and 51% with Z is conclusive evidence that Z split later than Y, since percentages of shared cognates are only at best a rough measure of time depths.

Some then will simply pick a maximum amount - a tolerance - by which two percentages may differ and still be considered to reflect the same time depth; tolerances of a few (usually two to five) percentage points are mostly used, the choice of a particular value being decided on an impressionistic view of the overall pattern of the table, many low percentages calling for a low tolerance, a majority of high percentages for a high tolerance.

Others, rather than hazarding an overall guess, will turn to a statistical formula which in effect does no more than calculate a tolerance for each new case encountered, and which has been the object of ample prescriptive and descriptive articles in linguistic publications (Gudschinsky, Simons).

The program was therefore written to allow its user to specify a tolerance as explained above or, if no tolerance was given, a confidence level.

SOME STATISTICAL NOTIONS EXPLAINED

Readers familiar with the notions involved may wish to skip the paragraphs which follow but which were called for by the frequent misconceptions found in linguistic publications on the subject (Simons, for instance, calls confidence level what is in fact the significance level and calls for a one-tailed test where a two-tailed test is appropriate).

Significance level vs confidence level (or coefficient)

You figure that your selection for the next race has a one-in-a-hundred chance of winning. The SIGNIFICANCE level of your selection losing the race is 0.01 (it will not SIGNIFY much to you if it does lose, since you do not really expect it to win). Conversely, the CONFIDENCE level for it losing the race is 0.99 (you are 99% CONFIDENT that it will lose). The terms CONFIDENCE INTERVAL and CONFIDENCE LIMITS are also used, the former being the one almost exclusively found in specialized literature.

Confidence levels in glottochronology

These notions creep into the glottochronologist's work whenever some language X shows suspiciously close percentages of shared cognates with two other languages, the question then being: are those

percentages close enough to each other to allow for the possibility that those languages did in fact split at the same time? To answer this question glottochronologists apply a mathematical formula involving the percentages under scrutiny, the number of items compared, and an arbitrarily-chosen level of confidence (or significance, one being the converse of the other); alas, it is generally not understood that the result of the computation is NOT a measure of the probability that the languages involved did in fact split at the same time, but only answers the question: "Knowing that I want to be 95% sure (0.95 level of confidence, or 0.05 level of significance) that no possible multiple splits will go undetected, could these percentages possibly reflect a multiple split?" In that case, the formula will yield answers which will trap 95% of all percentage pairs that COULD reflect such a split. This does NOT mean that 95% of the percentages thus trapped DO reflect such splits. It does however mean that if there are any such percentage pairs, then about 5% are likely to go undetected. The formula can be compared to a dragnet, the level of significance being the size of the mesh: as the level of significance is reduced (or the level of confidence increased), fewer and fewer percentage pairs that could reflect a three-way split are allowed to escape, but more and more percentage pairs that do not reflect any such splits are dragged in as well.

One-tailed vs two-tailed tests

The policeman who has you blow into a breathalyzer is only interested in finding out whether you are above the legal limit: he is performing a one-tailed test. The physician who takes your blood pressure wants to know not only if it is above the minimum safe, normal, or healthy level, but also below the maximum safe level: he is performing a two-tailed test. When the law is amended to make it also illegal to drive with LESS than a prescribed concentration of alcohol in the blood, policemen will be performing two-tailed tests.

The formula used by glottochronologists to determine whether two different percentages might reflect the same time depth is based on the calculation of a range within which those percentages should fall (i.e. a two-tailed test). The reference to one-tailed tests by some authors probably stems from the mistaken belief that it is based on the computation of a limit below which the difference between those two percentages should fall.

For a clear example of how to decide between one-tailed and two-tailed tests, see problem 10.22 in Spiegel 1972.

TESTING TRADITIONAL LEXICOSTATISTICAL TECHNIQUES

THE TESTS

The program was run six times, using a different combination of options every time. From its output ([10a] to [15a]) six trees were

drawn by hand ([10b] to [15b]):

1. Minimum-percentage method, no tolerance ([10a], [10b]).
2. Minimum-percentage method, tolerances calculated for a confidence level of 0.95 ([11a], [11b]). The figure of 0.95 was chosen, not because it has any intrinsic virtue, but because it is the one most frequently advocated in the literature on lexicostatistics and therefore most likely to be used by a linguist.
3. Maximum-percentage method, no tolerance ([12a], [12b]).
4. Maximum-percentage method, 0.95 confidence level ([13a], [13b]).
5. Mean-percentage method, no tolerance ([14a], [14b]).
6. Mean-percentage method, 0.95 confidence level ([15a], [15b]).

THEIR PERFORMANCE

Comparing the trees thus produced with the known tree of the language family shows the performance of those methods to range from poor (tolerances set to zero) to incredibly bad (tolerances computed for a level of confidence of 0.95). In all cases the MIKE-NOVEMBER split was grossly misplaced. There is little to pick between the results obtained by using zero tolerances; the trees produced by options 3 and 5 (maximum and mean percentage methods) are even surprisingly similar, the mean percentage method yielding more accurate time depths.

It will be argued that the wordlists were too small (40 items) and that the times between successive splits of a communalect were often too short (as little as 100 years in some cases). Quite so, and methods which, given the same set of data, would perform visibly better should therefore be given all the more consideration.

CHAPTER 3: METHODS BASED ON LINEAR-CORRELATION COEFFICIENTS

UNDERLYING PRINCIPLES

Imagine a language A splitting into communalects A1 and A2 ([16]). One thousand years later, A1 has retained 90% of its vocabulary whereas A2, for some reason or other, has retained only 30%. A1 and A2 are the only two surviving descendants of A, but have many 'cousins' (parent languages derived from an ancestor of A). The cognate percentages of A1 and A2 with their cousins will show a strong linear correlation: A1 will score on the average three times as much with any cousin as A2 will (you are urged to verify this statement by doing the calculation yourself). There will be of course variations in that 3 to 1 ratio; but, when plotted as in [4], pairs of percentages will tend to cluster along a straight line with a slope of 3.

> An algebraic quantity called linear correlation coefficient expresses how close to that line the points cluster. Its sign indicates in which direction the line slopes and its absolute value can vary from 1 (all points exactly on the line) down to 0 (no linear pattern at all). The basis for its computation is the square of the distance of each point to the line along which they all appear to cluster. The computational shortcut for finding the line that passes closest to all points, such that the sum of the squares of their distances to it is lower than it would be with any other straight line, is called linear regression.

Now let A2 split into a number of communalects A2a, A2b, etc., which start replacing vocabulary at different rates. Assuming that all survive, consider the situation after 1000 years ([17]). The scores of any of A2's descendants with its cousins (B, C, D, ... Z) will still tend to be in a constant ratio with the scores of A1's descendant with those same cousins, whatever their individual retention rates for the past 1000 years. If their scores with A1's descendant and any given descendant of A2 are again plotted as in [4], cousin languages will now again be represented by points closely clustered along a straight line, yielding a high, positive linear-correlation coefficient; if now plotted in the same manner onto the same diagram, the sister languages of A2a (A2b, A2c, etc.) will tend to be scattered off that main line or, if they happen to have replaced vocabulary at the same rate, they will cluster along a secondary line. Whichever the case, their addition to the graph obscures the single-line pattern and the linear-correlation coefficient drops accordingly (you are again urged to check this claim by filling in some of the unknowns in figure [17] and drawing the corresponding graph). The lower correlation between A1 and A2a is brought about by the fact that they no longer have a common

immediate ancestor. Thus high linear-correlation coefficients point to the most recent splits of all, whatever the individual past retention rates.

A negative correlation between two languages A and B means that the more cognates between A and any third language X, the fewer between B and X. Such a situation arises when A and B share no common lexical innovations and points to the earliest split in a language family. A positive correlation between A and B means that the more cognates between A and X, the more between B and X also. This situation arises when A and B do show common innovations, i.e. when some of the forms innovated by their closest common ancestor since the earliest split of the family have not been overwritten by later innovations. The correlation coefficients extracted from the cognate percentages of a language family provide therefore a measure of the amount of surviving common innovations since the first split of the family.

RECONSTRUCTING FAMILY TREES FROM CORRELATION COEFFICIENTS

A computer program was written to calculate correlation coefficients from cognate percentages and applied to table [9]. The resulting table ([18]) showed five chains of highly inter-correlated languages:

1) ALPHA-BRAVO-CHARLIE-DELTA
2) ECHO-FOXTROT-GOLF-HOTEL
3) INDIA-JULIET-KILO-LIMA
4) MIKE-NOVEMBER
5) OSCAR-PAPA-QUEBEC-ROMEO-SIERRA

The languages in the first chain showed markedly negative correlation coefficients with the rest, clearly reflecting the earliest split of the family into the two great subgroups ALPHA-DELTA and ECHO-SIERRA, but the picture was somewhat obscured by the fairly high correlation coefficients of LIMA with the individual members of chains 2, 4, and 5, so that no further subgrouping was immediately evident.

A minimal-spanning tree [19] was drawn from [18]. Bearing in mind that innovations by a common ancestor tend to be overwritten by later innovations, one can hold that the correlation of LIMA with KILO (0.945) suggests that they share roughly as many common innovations as FOXTROT does with GOLF (correlation: 0.946). Now we know from the record kept by the simulation program ([8]) that FOXTROT and GOLF's common ancestor had made seven innovations since the original split of the family, and LIMA and KILO's common ancestor six. A similar correlation is exhibited by PAPA and SIERRA (0.942), whose common ancestor had made six innovations, QUEBEC-SIERRA (0.944, six innovations), etc. There seemed to be some validity, then, in the method.

A 2-WAY SPLITTING ALGORITHM

Method

Consider the table of correlation coefficients [18]. The two languages with the lowest correlation (here BRAVO and MIKE, with a strong negative correlation of -0.856) must belong to two distinct groups. We therefore assign BRAVO to group#1 and MIKE to group#2. Of the remaining languages, the one which shows the highest correlation with either group#1 (represented by its single member BRAVO) or group#2 (represented by its single member MIKE) gets assigned to that group; ALPHA, with a correlation of 0.997 with BRAVO, is therefore assigned to group#1. The next most highly correlated language with either group is now CHARLIE with correlations of 0.988 and 0.991 respectively with ALPHA and BRAVO, the two current members of group#1, to which it therefore gets assigned.

In groping thus for a method for subgrouping languages automatically, we are now confronted with the problem of how to define the correlation of one language with a group of languages. There are three obvious choices, corresponding to the three main options offered by current lexicostatistical methods:

1. take the highest correlation (here that of CHARLIE with BRAVO, 0.991),
2. take the lowest correlation (CHARLIE with ALPHA, 0.988),
3. take the mean correlation (i.e. (0.988+0.991)/2 = 0.9895).

The difference between the three alternatives is here minimal but, as groups become larger, the gap between the lowest and highest correlations of the members of a group with an outsider widens considerably and it becomes imperative to stick to just one definition. Although it would have been possible to test all three alternatives on data produced by a large number of different simulations, I chose to follow my intuition and to define the correlation of a language with a language group as the mean of the correlations of that language with each language of the group.

There are now four languages assigned to either of the two groups: ALPHA, BRAVO, and CHARLIE to group#1, MIKE to group#2. Next to join a group is DELTA, with a correlation of (0.988+0.991+1.000)/3 = 0.993 with group#1. Once DELTA is incorporated into group#1, the highest correlation found with either group is that of NOVEMBER with group#2 (0.918 with MIKE, the only current member of group#2), and NOVEMBER is therefore assigned to group#2.

This process is continued until all languages have been assigned to a group. This done, the correlations of each language with its own group and with the opposite group are recomputed.

To understand why this is necessary, consider the case of NOVEMBER just after it was incorporated into group#2. Its correlation with group#2 was then equal to its correlation with

the only member of group#2, MIKE, i.e. 0.918. But once all languages are assigned to either group, the correlation of NOVEMBER with group#2, defined as the mean of its correlations with the members of that group, becomes much lower (0.629). Had its correlation with the opposite group been higher than 0.629 it should then have been removed from group#2 and reassigned to group#1.

This process of recalculating the correlations of each language with its own group and with the opposite group and of reassigning languages where necessary is repeated until the two groups are stable (just one iteration proved sufficient in all the cases later examined).

Each group thus obtained is then split into two using the same method, and this splitting process is continued until no further splitting is possible.

Options

A question comes to mind at this stage: the first splitting is done on a table of correlation coefficients calculated from a table of cognate percentages; for each of the two groups thus obtained is it not necessary then to recompute a correlation table? For, if we had been given, not the one cognate percentage table of the ALPHA-SIERRA family, but two percentage tables, one for ALPHA through to DELTA and the other for ECHO through to SIERRA, we would have computed two separate correlation tables, one from the cognate percentages of the ALPHA-DELTA family and one from the percentages of the ECHO-SIERRA family. It can be argued that by doing so one treats the two sets of cognate percentages as if they represented data from two unrelated language families by ignoring the existence of available cognate counts between them and thereby denies oneself potentially useful information. Possibly so, but the choice is here difficult.

Performance

A computer program was written which offered its user the choice of either option; its results are shown in diagrams [20] and [21].

Following each language name with its number in parentheses is a string of 1's and 2's indicating to which group or subgroup the language was assigned on successive splits. For instance ([20]), the first split assigned languages ALPHA through to DELTA to group#1, and the rest to group#2 (first column of digits). Group#2 was then split into two subgroups, MIKE and NOVEMBER being assigned to subgroup#2, the rest to subgroup#1 (second column of digits). The languages have been sorted by the program so that the corresponding tree can easily be drawn directly on the printout.

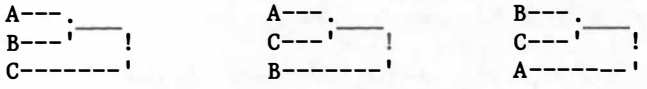
Comparing the trees thus reconstructed with the actual phylogenetic tree of the language family ([7]) shows [21] to be closer to the truth than [20]; bearing in mind that the lengths of the branches of the trees thus reconstructed bear no relationship to time depths, but that the relative positions of the nodes do, we see that [20] shows only one node (N) between the root (R) and the MIKE-NOVEMBER node (A), whereas [21] shows two nodes (M) and (N) between the MIKE-NOVEMBER node (A) and the root (R). Since nodes represent successive splits, the MIKE-NOVEMBER split is reconstructed in [21] as relatively late, and in [20] as relatively early. We know the former to be closer to the truth. Furthermore, [20] shows languages OSCAR through to SIERRA as having split quite late from the rest, whereas [21] shows this split as having occurred quite early. Here again, [21] is closer to the truth.

INTERPRETING BINARY TREES

Languages when they split do not necessarily split into just two, so that if at any stage of the evolution of a language family a multiple split occurred it can only be represented by two or more nodes in the binary tree reconstructed by a 2-way splitting algorithm: two successive nodes of the binary tree may in fact correspond to a single node of the true phylogenetic tree, or, in other words, any number of branches in the binary tree may be spurious.

Consider a 3-way split:

A 2-way splitting algorithm can only reconstruct such a split in one the following three ways:



Consider then again reconstruction [21]. Any of the branches of the tree could be spurious. Now remove the branch linking node (N) to node (O). The resulting tree is shown in [22]. The three-way split of the ECHO-SIERRA subgroup into the three subgroups ECHO-HOTEL, INDIA-NOVEMBER, and OSCAR-SIERRA, is now clearly visible. Try removing branches from tree [20] to show the same three-way split; it cannot be done. Tree [21] was, after all, a much truer reconstruction than it seemed. Most importantly, although absolute time depths are not provided by the algorithm, the placing of the MIKE-NOVEMBER split at a node far removed from the root points to a late split, which classic lexicostatistical methods failed to recognize.

SHORTCOMINGS OF 2-WAY SPLITTING ALGORITHMS

Given a binary-tree representation, one unfortunately cannot decide which branches are spurious and which are not, and an algorithm which would accurately reconstruct multiple splits as such rather than as successive 2-way splits would be even more valuable.

AN N-WAY SPLITTING ALGORITHM

Method

Consider the method followed in the 2-way splitting algorithm: the two most dissimilar languages become the nuclei of two embryonic groups, after which the remaining languages, one by one, come to join one group or the other. Thus, once the two most dissimilar languages are identified, the procedure followed changes from a splitting algorithm to a clustering algorithm. It should then be possible to split a group into an unspecified number of subgroups using a clustering algorithm, provided that some criterion of when to stop the clustering process could be found.

Examine the table of correlations [18]. The languages involved clearly fall into just two groups: ALPHA-DELTA, and ECHO-SIERRA. Observe that the correlations between any two languages of one group are strongly positive, those between any two languages of different groups just as strongly negative, so that it is perfectly obvious that a clustering algorithm should stop before merging those two groups. The signal to stop would be very much less obvious, however, if correlations were not all so strongly either negative or positive. Take for instance ([23]) the linear-correlation coefficients of twenty-five Austronesian languages computed from Tryon's cognate percentages (1978:891), where few coefficients are negative and most are rather unconvincingly positive: beyond the obvious - that the clustering should stop before Roviana and Sengga are joined to Xaraci and Nengone - little can be decided.

Consider again table [18], and examine the correlation coefficients within each of the two groups ALPHA-DELTA and ECHO-SIERRA. ALPHA-DELTA gives the impression of a much greater internal affinity than ECHO-SIERRA: the correlation coefficients between its members appear, on the average, much higher than within ECHO-SIERRA. It seems that the mean of the correlation coefficients of the languages of a putative group with each other somehow reflects the internal affinity, or cohesion, of that group. Note how, should ALPHA-DELTA be merged with ECHO-SIERRA, a large number of very low correlations would suddenly be brought in and how steeply the internal affinity of the new group would drop. But that very lowly correlated group would be none other than the language family which was being split. Could it be then that clustering should proceed only as long as the internal affinities of the groups being formed remain higher than that of the group being split?

Options

This new approach was translated into a computer program. Here again, as for the 2-way splitting algorithm, the user was given the option of having correlation coefficients recomputed between successive splits.

Performance

Diagrams [24] and [25] show the reconstructions obtained by that new method. Tree [25], obtained without recomputing separate correlation tables for each new group identified, accurately shows the earliest split of the family into ALPHA-DELTA and ECHO-SIERRA, the latter splitting into three subgroups: ECHO-HOTEL, INDIA-NOVEMBER, and OSCAR-SIERRA. Further splits, however, are inaccurately reconstructed; this is not surprising: as the groups to be split become smaller and smaller, the splitting algorithm has less and less information to work on within each group, and its decisions become accordingly increasingly inaccurate. Consequently, the splitting algorithms presented here are best suited for the reconstruction of the earliest history of a language family.

CHAPTER 4: METHODS BASED ON THE DIRECT EXAMINATION OF THE WORDLISTS

One may wonder at this stage whether the amount of information discarded in the computation of cognate percentages and of derived measurements such as linear-correlation coefficients is not such that the accuracy of the reconstructions can only be gravely impaired, and perhaps methods based on the examination of the distribution of the individual cognate groups would yield such vastly more accurate reconstructions that the additional computational cost involved would be well justified.

THE REDUCED MUTATION ALGORITHM

METHOD

Hartigan (1975:237-241) describes a clustering algorithm which he then goes on to illustrate by applying it to the amino-acid sequences in a protein molecule of six species of vertebrates (man, monkey, chicken, duck, kangaroo, and rattlesnake). The algorithm reconstructs a sequence of past mutations from which a phylogenetic tree can be drawn (Hartigan 1975:241). The analogy between vocabulary replacement and mutations (replacements of parts of the genetic message) is striking, so much so that Hartigan gives a list of words in 13 Indo-European languages (supplied by Dyen) as an example of the type of data to which that clustering method can be validly applied.

IMPLEMENTATION

The algorithm was translated into a program which was carefully debugged and checked.

> It was first tested on the very set of amino-acid data given by Hartigan, but gave results different from Hartigan's, suggesting a programming error on my part. The calculations were then carried out by hand, and the results obtained at each step compared with a printout of the intermediate results given by the program. It appeared that the discrepancy was due, not to a bug in the program, but to Hartigan's miscalculation of the distance between the amino-acid sequence of the kangaroo and that of the, earlier merged, chicken-duck cluster. Most probably, Hartigan did not use a computer program to illustrate the use of the reduced-mutation algorithm, but did the calculations by hand. Whatever the reason for the error, the computer-produced tree gave a more satisfactory reconstruction, as it grouped the kangaroo with the

man-monkey cluster rather than the chicken-duck cluster as in Hartigan's book.

PERFORMANCE

The program was fed the wordlists of the simulated language family, and a phylogenetic tree ([26]) drawn from the account of the successive mergings of lists and of the predicted past individual word replacements. The tree thus reconstructed is strikingly similar to tree [12b], obtained by traditional lexicostatistical techniques using the mean-percentage method and a zero tolerance.

As implemented, the reduced mutation algorithm was extremely slow, requiring about 120 seconds of CPU time on a DEC-KL10, whereas none of the other methods described so far had taken more than 0.5 seconds to process the percentage table, which had been produced from the wordlists in just 0.4 seconds. Admittedly, the reduced-mutation program had been written with readability and ease of debugging in mind rather than efficiency and could have been optimized. But since its performance was indistinguishable from that of the traditional methods, there was no point in trying to improve its speed of execution, as it was unlikely to be put to extensive use.

THE WILD-CARD CLUSTERING ALGORITHM

METHOD

Consider the first 25 items in the wordlists of ALPHA through to SIERRA, reduced to cognate groups:

ALPHA	1	0	2	0	1	1	0	1	2	1	4	1	0	0	0	0	1	2	2	0	1	2	0	1	0
BRAVO	1	0	4	0	1	2	0	1	2	1	4	1	0	0	0	2	1	2	2	0	1	2	0	1	0
CHARLIE	1	1	1	0	1	2	0	1	2	1	4	1	0	0	0	2	1	2	2	0	1	2	0	1	0
DELTA	1	1	1	0	1	2	0	1	2	1	4	1	0	0	0	2	1	2	2	0	1	2	0	1	0
ECHO	1	2	6	1	1	2	0	3	4	1	0	0	3	0	2	2	0	1	2	0	1	2	0	2	0
FOXTROT	1	0	6	1	1	2	0	0	1	1	0	0	3	0	2	2	0	1	0	2	1	4	0	0	0
GOLF	1	0	6	1	2	2	0	0	5	0	0	0	3	3	2	2	0	1	2	0	1	2	0	0	0
HOTEL	1	0	5	1	2	2	0	0	5	1	0	0	3	3	2	2	2	1	2	0	1	2	0	0	0
INDIA	1	0	0	1	1	2	0	1	5	1	0	1	0	1	2	2	0	0	2	3	1	3	0	3	1
JULIET	1	0	6	1	1	2	3	1	5	1	0	1	2	0	2	3	0	1	2	3	1	3	0	3	1
KILO	0	0	6	1	1	2	0	1	5	1	0	1	0	0	2	2	0	1	2	0	1	3	0	3	1
LIMA	1	0	6	1	1	2	1	4	3	1	0	1	1	0	2	4	0	1	2	0	0	3	0	0	4
MIKE	1	0	3	1	1	2	2	2	5	1	1	1	0	0	2	1	0	1	1	1	1	0	0	0	3
NOVEMBER	1	0	6	1	0	0	0	1	5	1	2	3	0	0	2	2	0	1	2	1	1	0	0	0	3
OSCAR	1	0	6	1	1	2	0	1	5	1	0	1	0	2	2	5	0	1	2	0	1	2	0	0	2
PAPA	1	0	6	1	1	2	0	1	5	1	0	1	0	0	2	5	0	1	2	0	1	2	0	0	0
QUEBEC	1	0	6	1	1	2	0	1	5	1	0	2	0	0	2	5	0	1	2	0	1	2	0	0	0
ROMEO	1	0	6	1	1	2	0	1	0	1	3	1	0	0	2	5	0	1	2	0	1	2	1	0	5
SIERRA	1	0	6	1	1	2	0	1	5	1	0	1	0	0	1	5	0	1	2	0	2	1	0	0	0

ECHO is the only list which has "2" for item#2, and CHARLIE and DELTA are the only two lists which have "1".

It is unlikely that CHARLIE and DELTA should have independently replaced item#2, which is almost everywhere else represented by the same cognate ("0"). They most probably inherited "1" from a common ancestor.

Now ECHO is unique in having "2" for that same item. Since it seems quite unlikely that ECHO should have been the only language to have retained the original form, "2" was very probably innovated by ECHO. We cannot tell what the form inherited by ECHO from its immediate ancestor was, and we therefore cannot hold that ECHO's item#2 was NOT represented in its immediate ancestor's wordlist by "1" or "0", found in the other languages. The unique form displayed by ECHO for item#2 is then potentially cognate with any other form: a wild card. Further examining ECHO's list we find three more such wild cards: items#8 ("3"), #9 ("4"), and #24 ("2"). Neither CHARLIE nor DELTA shows such unique forms for any item, and their wordlists therefore contain no wild cards. ALPHA shows three (items #3, #6, and #16), and BRAVO one (item#3). Thus, with asterisks symbolizing wild cards:

```
ALPHA       1 0 * 0 1 * 0 1 2 1 4 1 0 0 0 * 1 2 2 0 1 2 0 1 0
BRAVO       1 0 * 0 1 2 0 1 2 1 4 1 0 0 0 2 1 2 2 0 1 2 0 1 0
CHARLIE     1 1 1 0 1 2 0 1 2 1 4 1 0 0 0 2 1 2 2 0 1 2 0 1 0
DELTA       1 1 1 0 1 2 0 1 2 1 4 1 0 0 0 2 1 2 2 0 1 2 0 1 0
ECHO        1 * 6 1 1 2 0 * * 1 0 0 3 0 2 2 0 1 2 0 1 2 0 * 0
.....       . . . . . . . . . . . . . . . . . . . . . . . . .
```

Compute the proportion of shared items between ALPHA, BRAVO, CHARLIE, etc., with wild cards always scoring a match:

```
ALPHA
1.000  BRAVO
0.960  0.960  CHARLIE
0.960  0.960  1.000  DELTA
0.720  0.720  0.640  0.640  ECHO
.....  .....  .....  .....  .....  .....
```

Lists which share 100% items are now merged: ALPHA with BRAVO, CHARLIE with DELTA, etc. When two forms are merged, one of which is a wild card, the natural replaces the wild card. Thus:

```
ALPHA-BRAVO     1 0 * 0 1 2 0 1 2 1 4 1 0 0 0 2 1 2 2 0 1 2 0 1 0
CHARLIE-DELTA   1 1 1 0 1 2 0 1 2 1 4 1 0 0 0 2 1 2 2 0 1 2 0 1 0
ECHO            1 * 6 1 1 2 0 * * 1 0 0 3 0 2 2 0 1 2 0 1 2 0 * 0
.......         . . . . . . . . . . . . . . . . . . . . . . . . .
```

Forms unique to a language or mesolanguage now become wild, and the process is repeated until all lists have been merged into one.

LIMITATIONS

Consider two languages or mesolanguages X and Y, each showing a unique, i.e. wild-card, form for a certain item. It is impossible to ascertain whether

1. X and Y independently replaced that item,

2. their immediate ancestor having replaced the item in question both X and Y replaced it again,

3. their immediate ancestor having replaced that item, either X or Y, but not both, replaced the item again.

Case 2 (the same item independently replaced three times) is the least probable of all three but cases 1 and 3 are equally likely (the same item replaced twice). There is no way to deicde which one of these events did in fact take place, and absolutely accurate reconstructions are therefore not normally possible (the program always opted for the first case).

The more mergings are carried out the more such situations are likely to arise. A wrong choice inevitably leading to a slightly wrong reconstruction of the mesolanguage's wordlist, the information on which the algorithm works becomes increasingly inaccurate as earlier and earlier states are reconstructed; the wild-card method, then, is best suited for the reconstruction of the most recent history of language families.

It would have been possible to modify the wild-card program so that, whenever faced with a such a choice, it would investigate the consequences of all three options, as some chess programs investigate the consequences of all legal moves up to a variable number of moves ahead. As implemented, the wild-card algorithm took only 1 second of CPU time to process the wordlists of the 19 languages of the simulation, and, very probably, could have been made to explore all possible alternative branchings within a reasonable amount of time.

Given a larger number of longer wordlists, however, not only does the time spent reconstituting one particular tree increase, but the number of possible branchings, therefore of alternatives to be explored when confronted with the kind of choice described above, increases at such an immensely fast rate that a modified program could be expected to require hours of computer time to process moderately large amounts of data.

Furthermore, the vexing question would have remained of which reconstruction was the valid one. If, in accordance with Occam's razor, the reconstruction involving the least number of word replacements were retained, then tree [27], which involves 121 replacements, would have to be preferred to the true phylogenetic tree, which involves 122 replacements. There was therefore no point in modifying the program to produce alternative reconstructions.

PERFORMANCE

Detailed inspection of the results

The computer program was fed the wordlists of languages ALPHA through to SIERRA. Its output is given hereunder, with a running commentary. The phylogenetic tree drawn from the information in this output is shown in [27].

Group#1 = ALPHA, BRAVO
10.012012141000212201201000420010121 1030
The first mesolanguage identified was reconstituted from ALPHA and BRAVO. Its wordlist is given with a dot representing a wild card form (hence unknown at this stage of the reconstruction).
ALPHA replaced 3 items: 3 6 16
10*01*012141000*12201201000420010121 1030
ALPHA replaced items#3, 6, and 16 on the wordlist it had inherited from the mesolanguage from which it issued. Its wordlist is then given, with an asterisk in the position of each item it replaced. A check with the record of the derivation of the language family shows this to be correct.
BRAVO replaced 1 item: 3
10*012012141000212201201000420010121 1030
Also correct.

Group#2 = CHARLIE, DELTA
111012012141000212201201000420010111 1030
CHARLIE replaced 0 items:
111012012141000212201201000420010111 1030
DELTA replaced 0 items:
111012012141000212201201000420010111 1030
Correct.

Group#3 = ECHO, FOXTROT
10611200.1003022012012000003520103203 0.2
ECHO replaced 7 items: 2 8 9 24 31 37 39
1*61120**10030220120120*000352*10320*0*2
FOXTROT replaced 9 items: 9 19 20 22 27 28 35 39 40
10611200*100302201**1*0000**520103*030**
Correct.

Group#4 = GOLF, HOTEL
106122005100332201201200000352010340 3002
GOLF replaced 3 items: 10 29 32
106122005*00332201201200000 3*20*03403002
HOTEL replaced 2 items: 3 17
10*1220051003322*120120000035201034 03002
Correct.

Group#5 = INDIA, JULIET
106112015101002201231303100352010322 3202

```
INDIA replaced 6 items:  3 14 18 26 32 33
10*1120151010*220*2313031*03520**3223202
JULIET replaced 3 items:  7 13 16
106112*15101*02*012313031003520103223202
  Correct.

Group#6 = MIKE, NOVEMBER
1061120151.10022012110003203530103203001
MIKE replaced 7 items:  3  7  8 11 16 19 32
10*112**51*1002*01*110003203530*03203001
NOVEMBER replaced 6 items:  5  6 11 12 29 35
1061**0151**0022012110003203*30103*03001
  Correct.

Group#7 = PAPA, QUEBEC
10611201510100250120120000035141223000002
PAPA replaced 2 items: 29 38
1061120151010025012012000003*14122300*02
QUEBEC replaced 1 item: 12
10611201510*00250120120000035141223000002
  Correct.

Group#8 = Group#1, Group#2
10.012012141000212201201000420010121 1030
Group#1 replaced 0 items:
10.012012141000212201201000420010121 1030
Group#2 replaced 2 items:  2 35
1**012012141000212201201000420010 1*11030
  Correct: Group#1 is ALPHA-BRAVO, Group#2 mesolanguage CD.  To say
  that  Group#1 made no innovations is to say that either ALPHA and
  BRAVO  split  directly  from  the  mesolanguage  represented  by
  Group#8,  or  that  their  immediate  ancestor  (Group#1) had not
  replaced  any  words  on  the  wordlist  inherited  from  its  own
  immediate ancestor (Group#8).

Group#9 = Group#3, Group#4
10611200510030220120120000035201032030 02
Group#3 replaced 0 items:
10611200.10030220120120000035201032030.2
Group#4 replaced 3 items:  5 14 35
1061*20051003*2201201200000352010 3*03002
  Correct: Group#3 is mesolanguage EF, Group#4 mesolanguage GH.  GH
  did replace items#5, 14, and 35.

Group#10 = Group#5, KILO
10611201510100220120130310035201032 3.02
Group#5 replaced 3 items: 20 36 38
1061120151010022012*130310035201032*3*02
KILO replaced 5 items:  1 28 31 37 38
*0611201510100220120130310 0*52*10320**02
  Slightly incorrect: Group#5 is mesolanguage IJ from  which  INDIA
  and  JULIET issued.  IJ did replace items 20 and 36, but not item
```

38. KILO's innovations are correctly reconstructed.

```
Group#11 = LIMA, Group#6
1061120151010022012013000.003520103203002
LIMA replaced 10 items:  7  8  9 13 16 21 25 31 34 38
106112***101*02*0120*300*00352*10*203*02
Group#6 replaced 6 items: 20 22 25 26 30 40
1061120151.10022012*1*00**035*010320300*
```

Slightly incorrect: Group#6 is mesolanguage MN (MIKE-NOVEMBER). Its replacements are accurately predicted; LIMA, however, did not replace item#25.

```
Group#12 = OSCAR, Group#7
10611201510100250120120000035241223000002
OSCAR replaced 2 items: 14 25
1061120151010*2501201200*003524122300002
Group#7 replaced 1 item: 30
106112015101002501201200000 35*4122300002
```

Correct: Group#12 is mesolanguage PQ (PAPA-QUEBEC). Both PQ's and OSCAR's innovations are correctly identified.

```
Group#13 = Group#8, Group#9
106112015101002201201200000352010320 3002
Group#8 replaced 15 items:  4  9 11 15 17 18 24 28 29 30 34 36 37 39
40
10.*1201*1*100*2**20120*000***010*2**0**
Group#9 replaced 3 items:  8 12 13
1061120*510**022012012000003520103203002
```

Wrong: Group#8 is mesolanguage AD, Group#9 mesolanguage EH. The branching here is wrong. Moreover, AD replaced only 12 of the 15 items listed here, and replaced item#3 which is not listed here. We know that AD had replaced item#3 by 1500 and that ALPHA and BRAVO had replaced that same item again by 1900. The algorithm was unable to track down those three separate replacements of the same item. The items replaced by mesolanguage EH are correctly identified.

```
Group#14 = Group#10, Group#11
1061120151010022012013000*003520103203002
Group#10 replaced 1 item: 24
106112015101002201201 30**003520103203.02
Group#11 replaced 0 items:
1061120151010022012013000.003520103203002
```

Wrong: Group#10 is mesolanguage IK, Group#11 mesolanguage LN. The branching is incorrectly reconstructed, and IK replaced item#28 as well as item#24.

```
Group#15 = Group#12, ROMEO
10611201510100250120120000035241033000 02
Group#12 replaced 2 items: 33 34
1061120151010025012012000003524 1**300002
```

ROMEO replaced 6 items: 9 11 23 25 28 36
10611201*1*10025012012*0*00*5241033*0002
Correct: Group#12 is mesolanguage OQ. Branching and replacements are accurately predicted.

Group#16 = Group#13, Group#14
1061120151010022012012000003520103203002
Group#13 replaced 0 items:
1061120151010022012012000003520103203002
Group#14 replaced 1 item: 22
106112015101002201201*00.003520103203002
The branching of Group#13 and Group#14 was wrong, and so is their merging here into the same cluster.

Group#17 = Group#15, SIERRA
1061120151010025012012000003520103200002
Group#15 replaced 2 items: 31 35
106112015101002501201200000352*103*00002
SIERRA replaced 5 items: 15 21 22 29 34
10611201510100*50120**000003*2010*200002
Correct: Group#15 is mesolanguage OR. Branching and replacements are accurate.

Group#18 = Group#16, Group#17
106112015101002.012012000003520103 20.002
Group#16 replaced 2 items: 16 37
106112015101002*012012000003520103 20*002
Group#17 replaced 2 items: 16 37
106112015101002*012012000003520103 20*002
Wrong again.

Conclusion

As pointed out above, the performance of the wild-card algorithm should be assessed on its restitution of the late history of the evolution of language families; judged on the reconstruction of the four great subgroups ALPHA-DELTA, ECHO-HOTEL, INDIA-NOVEMBER, and OSCAR-SIERRA, it can only be described as amazingly accurate: the internal reconstruction of the three smaller subgroups is perfect, that of the largest subgroup (INDIA-NOVEMBER) only slightly inaccurate. Its accurate placing of the MIKE-NOVEMBER split, where all traditional methods and the reduced mutation algorithm had failed, is especially noteworthy.

MISIDENTIFIED COGNATES

Configurations may arise which make further mergings impossible, e.g.:

```
ALPHA      0 0 . . . . . .
BRAVO      0 1 . . . . . .
CHARLIE    1 0 . . . . . .
DELTA      1 1 . . . . . .
.....      . . . . . .
```

The first two items show no unique forms (wild cards), and whatever the other forms in the lists, it will be impossible to merge them, as whichever pair is contemplated for merging, its two members will necessarily differ by at least one item. Such configurations cannot arise unless forms have been wrongly scored as cognates, being either loanwords wrongly identified as cognate forms or independently innovated forms which happen to resemble a possible cognate form.

> Assume that all loanwords are identified and scored as non-cognates, and that no case of an innovation resembling an already existing form may ever occur. Consider a language X splitting into a number of descendants A, B, C, etc. Any items they might innovate, until the time they split, are necessarily wild cards, for, on the above premises, innovations are independent, unmistakable, and therefore show up as unique forms, i.e. wild cards. Then, if at any time of the reconstruction the lack of wild cards prevents further mergings of the wordlists, at least one premise must be false: borrowings have been scored as cognates, or the data contained spurious cognate forms.

To a certain extent then, the wild-card algorithm also provides a check of whether the data has been correctly interpreted by its human user.

CHAPTER 5: ON SCORING LOANWORDS

Traditional lexicostatistical methods require that loanwords be scored as non-cognates. This position is hardly tenable under any model of language evolution. All the methods discussed so far were designed to process a particular type of information: messages (wordlists) randomly selected parts of which are subject to sudden, purely random changes (and especially, uninfluenced by the contents of other messages). In the case of borrowing, the selection of the particular items borrowed may indeed be random, but the resulting change in the affected part of the message is in no way random since it is a mere copy of the corresponding parts of some other message. Loanwords, then, do not constitute any kind of information in the terms of the evolutionary model which underlies those methods: they are noise (as meant in information theory). To score them as non-cognates (or as cognates for that matter) is like letting scratches on a record influence one's appreciation of the performance of a musical work. Loanwords should be treated as missing items, for that it precisely what they are, as scratches on a record are bits missing from the original pressing.

CHAPTER 6: EIGHT VARIATIONS ON THE ORIGINAL EXPERIMENT

THE NEED FOR FURTHER EXPERIMENTS

One of the splitting algorithms tried, and not mentioned in the previous chapters, was an adaptation of Sukhotin's algorithm for recognizing vowel symbols in an unknown alphabetical script. It performed perfectly on the language family created in the preliminary experiment, but failed even to identify the basic ALPHA-DELTA, ECHO-SIERRA dichotomy when applied to the data produced by further simulations of the evolution of the same language family (remember that the details of the evolution are not predetermined, so that, given the same guidelines - i.e. family tree, wordlist size, retention rate, and accuracy of cognate recognition, each successive simulation is likely to produce a different set of data). Its original success was a mere coincidence.

Not only must experiments be repeatable, they must be repeated as well.

THE EXPERIMENTS

The same family tree was used as in the original experiment, but with time depths of 2000 and 4000 years instead of 1900. The retention rate was set to 81% per 1000 years, and the accuracy of cognate recognition to the maximum allowable value, 9999. Four experiments were carry out with a time depth of 2000 years, four with 4000 years. Out of each set of four experiments, two were conducted with 50-item wordlists, and two with 200-item wordlists.

For each experiment six trees were drawn from the output of the program implementing traditional lexicostatistical techniques, two trees from that of the n-way splitting algorithm (one for each option), one tree from that of the reduced mutation algorithm, and one from that of the wild-card algorithm. The tree of the actual family created was drawn from the running account kept by the simulation program, and the tables of cognate percentages and linear-correlation coefficients were printed out. Thus thirteen tables and diagrams were prepared from the results of each experiment.

All 104 tables and diagrams are given in appendix I, each identified by a 3-digit number.

The first digit (1 to 8) is the number of the experiment:
1 and 2: 2000-year time depth, 50-item wordlists,
3 and 4: 2000-year time depth, 200-item wordlists,
5 and 6: 4000-year time depth, 50-item wordlists,
7 and 8: 4000-year time depth, 200-item wordlists.

The second digit identifies the program from the output of which the table or diagram was obtained:

0: tree drawn from the running account of the simulation program,
1: cognate percentages,
2: traditional lexicostatistical methods,
3: linear-correlation coefficients,
4: n-way splitting algorithm,
5: reduced mutation algorithm,
6: wild-card algorithm.

The third digit identifies the option used:

0: no option available,
1: minimum-percentage method, no tolerance,
2: minimum-percentage method, 0.5 confidence level,
3: mean-percentage method, no tolerance,
4: mean-percentage method, 0.5 confidence level,
5: maximum-percentage method, no tolerance,
6: maximum-percentage method, 0.5 confidence level,
7: correlations recomputed between successive splits,
8: correlations not recomputed.

The 0.5 confidence level for the traditional lexicostatistic methods was chosen over the more widely advocated 0.95 level as it proved to yield more meaningful, less undifferentiated results.

HOW THE VARIOUS METHODS PERFORMED

TRADITIONAL LEXICOSTATISTICAL TECHNIQUES

Minimum-percentage method

The basic binary split of the family into ALPHA-DELTA and ECHO-SIERRA was recognized in all cases, except in the second experiment (2000-year time depth, 50-item wordlists) when a confidence level of 0.5 was used.

The ternary split of ECHO-SIERRA into ECHO-HOTEL, INDIA-NOVEMBER, and OSCAR-SIERRA was recognized as such in experiments:

1: 2000 years, 50 items, either option,
4: 2000 years, 200 items, no tolerance,
7: 4000 years, 200 items, 0.5 confidence level.

It was identified as two successive binary splits in experiments:

7: 4000 years, 200 items, no tolerance,
8: 4000 years, 200 items, no tolerance.

In all other cases, i.e. 10 reconstructions out of 16, either no ternary split was recognized, or a ternary split was reconstructed but one or more languages were assigned to the wrong subgroup.

Mean-percentage method

The basic binary split ALPHA-DELTA and ECHO-SIERRA was recognized in all cases, except in experiment 5 (4000 years, 50 items, 0.5 confidence level). The ternary split of ECHO-SIERRA was reconstructed as such in experiments:

1: 2000 years, 50 items, 0.5 confidence level,
7: 4000 years, 200 items, 0.5 confidence level,
8: 4000 years, 200 items, 0.5 confidence level,

and as two successive binary splits in experiments:

2: 2000 years, 50 items, no tolerance,
4: 2000 years, 200 items, no tolerance,
7: 4000 years, 200 items, no tolerance,
8: 4000 years, 200 items, not tolerance.

The mean-percentage method thus failed in 9 reconstructions out of 16.

Maximum-percentage method

The basic binary split was recognized in all cases, and the subsequent ternary split of ECHO-SIERRA was reconstructed as such in experiment 7 (4000 years, 200 items, 0.5 confidence level) and as two successive binary splits in experiments:

4: 2000 years, 200 items, no tolerance,
7: 4000 years, 200 items, no tolerance,
8: 4000 years, 200 items, no tolerance.

The maximum-percentage method thus failed in 12 reconstructions out of 16.

Discussion

All eight experiments provided ideal conditions for traditional lexicostatistical techniques to perform to their best, as their two basic assumptions - a universal, constant retention rate and no borrowing - were respected by the computer simulation. And yet the reconstructions obtained were far from reliable, being grossly inaccurate in more than half the cases, even with the best method (mean percentages). How can it be that methods tested under ideal conditions should yield less than nearly perfect results? To reconstruct the family trees of related languages traditional lexicostatistics relies on the calculation of the probable times at which communalects might have split. This calculation is in turn based upon an assumed universal rate of vocabulary retention. The shorter the wordlists used, the less accurately this retention rate will manifest itself: roll a die once, twice, three, ten times and you will never be able to estimate the probability of throwing an ace; roll it 60 times and you get a rough idea, 120 times and you might feel confident enough to accept odds of 10 to 1, 6000 times and you will even accept any odds longer than 5 to 1. There can be no doubt that,

given long enough wordlists (2000 items perhaps), the traditional methods would have performed brilliantly. But, even though there were a universal, constant retention rate, of what practical use could possibly be methods requiring the collection of wordlists thousands of items long?

METHODS BASED ON LINEAR-CORRELATION COEFFICIENTS

Performance

The n-way splitting algorithm, applied to the tables of linear-correlation coefficients, recognized the basic binary split of the language family in all cases. Its success in identifying the subsequent ternary split of ECHO-SIERRA depended mainly on whether the correlation tables were recomputed afresh between successive splits.

When they were recomputed this ternary split was reconstructed as such in experiments:

1: 2000 years, 50 items,
4: 2000 years, 200 items,
5: 4000 years, 50 items,
7: 4000 years, 200 items,
8: 4000 years, 200 items,

and in no case was it reconstructed as two successive binary splits.

When correlations were not recomputed, the ternary split was reconstructed as such in experiments:

1: 2000 years, 50 items,
5: 4000 years, 200 items,
7: 4000 years, 200 items,
8: 4000 years, 200 items,

and as two successive binary splits in experiments:

2: 2000 years, 50 items,
3: 2000 years, 200 items,
4: 2000 years, 200 items.

Methods based on correlation coefficients thus failed in 3 cases out of 8 (correlations recomputed) and 1 case out of 8 (correlations not recomputed).

Discussion

The performance of these methods is far from perfect, and one may wonder if the splitting algorithm should not be blamed, as it often reconstructed the 3-way split of ECHO-SIERRA as two successive binary splits (experiments 2, 3, and 4). Upon examining the corresponding tables of correlation coefficients ([230], [330], [430]), however, it appears that the ECHO-SIERRA cluster could hardly have been divided otherwise.

The failure of both methods in experiment 6 is particularly distressing and deserves further attention. Here, MIKE and NOVEMBER

were wrongly assigned to the OSCAR-SIERRA cluster and LIMA to ECHO-HOTEL. Table [630] shows the correlations of MIKE and NOVEMBER to be indeed overall higher with languages OSCAR to SIERRA than with the rest, and the correlation of LIMA with the ECHO-HOTEL cluster higher than with any other. Here again then, the fault does not lie with the splitting algorithm. The choice of the correlation coefficient is probably to blame.

The correlation coefficients used were obtained by the standard linear-regression technique, which assumes a linear function of the form

$$y = ax + b$$

Thus, if two languages A and B share the following percentages of cognates with languages C to Q

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
A	1	3	4	4	7	5	8	8	9	10	6	2	4	0	6
B	52	56	58	58	64	60	66	66	68	70	62	54	58	50	62

since given the score x of a language A with any language C to Q, the score y of language B with that same language can be predicted EXACTLY from the formula

$$y = 2x + 50$$

the single-line pattern is perfect, the linear-correlation coefficient of A with B is therefore 1, the maximum possible value. This is counter-intuitive; and indeed, from the explanations given in chapter 3, one can see that the linear function which applies in the case of language evolution is of the form

$$y = ax$$

assuming that the retention rate is the same for all the items of a particular list at a particular time. The measure of correlation given by the standard linear-regression technique is therefore inappropriate, even though it does provide reconstructions more reliable than any of the traditional lexicostatistical methods. The elaboration of better measures of correlation will be treated in a forthcoming monograph.

Methods based on correlation coefficients were primarily designed to circumvent the distortions in percentage scores brought about by unequal vocabulary retention rates, and one may rightly wonder how such methods should have proved more reliable than traditional lexicostatistics in experiments where the retention rate was kept constant in time and equal for all languages.

A cognate percentage relies on data from two wordlists, a correlation coefficient between two of a group of n languages relies on n-2 cognate percentages, and therefore on data from 2(n-2) wordlists. The larger the number of wordlists, the more data enter into the computation of the correlation coefficient of each pair of

languages, and the more reliable the measure of their similarity is. This is very probably the reason why the method whereby correlation coefficients were computed afresh for each newly identified subgroup was less reliable, as those coefficients were obtained on smaller and smaller amounts of data.

REDUCED MUTATION ALGORITHM

Performance

The reduced mutation algorithm identified the basic binary split in all experiments, but did not succeed, even once, in reconstructing the subsequent ternary split of ECHO-SIERRA, either as such, or as two successive binary splits.

Discussion

The reasons for the resounding failure of the reduced mutation algorithm are somewhat akin to those for the failure of the traditional lexicostatistical method: the measure of the similarity or of the distance between two languages is based on data from just two wordlists. The measure of distance used by the reduced mutation algorithm is furthermore not reconciliable, at least in my eyes, with the linguistic model. Interested readers should refer to Hartigan 1975:233-246.

WILD-CARD ALGORITHM

Performance

It has been pointed out earlier that by its very nature the wild-card algorithm becomes more and more inaccurate as it attempts earlier and earlier reconstructions. And, not unexpectedly, it consistently failed to identify the basic binary split of the ALPHA-SIERRA language family. Once allowances have been made for this failure, the reconstructions are found to reflect the ternary split of ECHO-SIERRA as such in experiments:

2: 2000 years, 50 items,
3: 2000 years, 200 items,
5: 4000 years, 50 items,
7: 4000 years, 200 items,

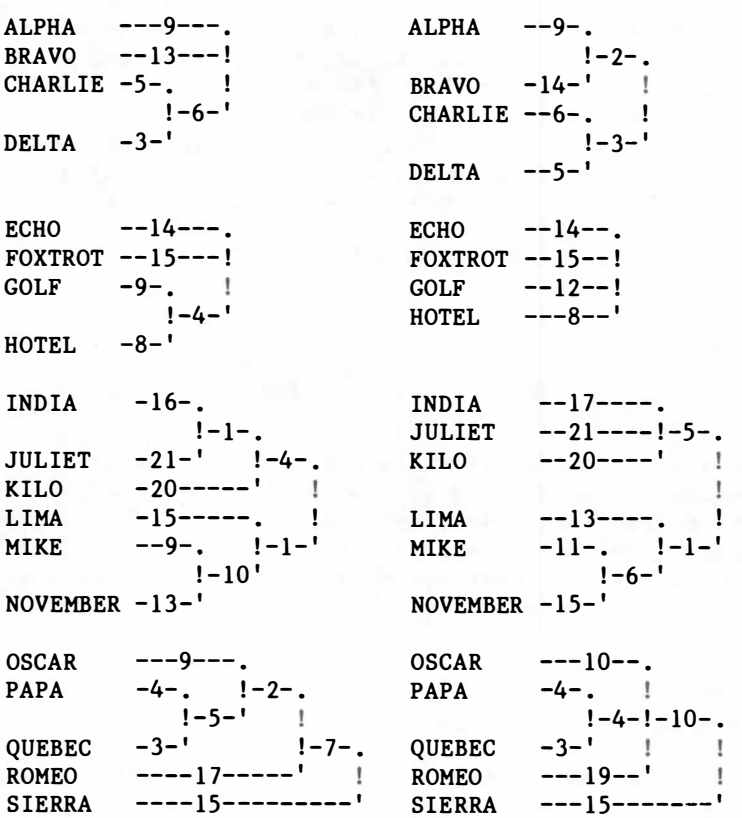
and as two successive binary splits in experiment 8 (4000 years, 200 items). Under the criteria used for judging the performance of the other methods, the wild-card algorithm failed in 3 cases out of 8, thus proving superior to traditional lexicostatistics but not to correlation-based methods.

Discussion

The performance of the wild-card algorithm might well be better than it appears:

In the fourth experiment the reconstruction was stopped by accidental cognate-like forms before ECHO-FOXTROT and GOLF-HOTEL could be merged, and this was counted as a failure. Had no spurious cognate forms been present, the reconstruction might have been successful.

In the first and sixth experiments ECHO-SIERRA is reconstructed as having undergone a 4-way split. There are no cases, however, where languages merged into a group have amongst them a language that rightly belongs to another group, and the number of innovations reconstructed by the algorithm as having occurred within each of the four great clusters ALPHA-DELTA, ECHO-HOTEL, INDIA-NOVEMBER, and OSCAR-SIERRA taken individually remains very close to the truth, even in the worst cases. Take for instance the tree ([660]) reconstructed in the sixth experiment and compare it subgroup by subgroup with the true tree ([600]), i.e. ALPHA-DELTA in [660] against ALPHA-DELTA in [600], ECHO-HOTEL in [660] against ECHO-HOTEL in [600], etc. Thus, with the number of items replaced shown on the branches:



It should be noted that it is in this very experiment that all other methods, whether traditional or correlation-based, failed to assign the proper languages to the three great subgroups issued from ECHO-SIERRA. The reconstruction provided by the wild-card algorithm is then all the more remarkable.

The wild-card algorithm is not based upon any assumptions about retention rates; it is unable to proceed when cognates have been misidentified; it is relatively inexpensive in computer time (the nineteen 200-item wordlists took under 10 seconds of CPU time on DEC-KL10, whereas the computation of the corresponding cognate percentage table took about 1.2 seconds and its interpretation a further 0.65 seconds, be it by a traditional lexicostatistical method or a correlation-based method); although on its own it cannot reconstruct the earliest splits of a language family, it can be used to reconstruct the history of subgroups already identified by other methods; and it allows the reconstruction not only of phylogenetic trees but also of the vocabularies of putative earlier languages with a fair degree of accuracy.

The reason for the fairly satisfactory performance of the wild-card algorithm seems, again, to stem from the fact that it takes into account data from more than just two wordlists at a time, as wild-card forms are identified by examining all extant wordlists.

Variations and adaptations of the wild-card algorithm will be investigated in a later monograph.

CHAPTER 7: CONCLUSION

TRADITIONAL LEXICOSTATISTICAL METHODS

PREREQUISITES FOR THEIR USE

Traditional lexicostatistical methods have proved rather unreliable even under ideal conditions. Linguists wishing to apply them to their data should be warned that totally misleading reconstructions may be arrived at under less than nearly ideal conditions, i.e.

1. When vocabulary retention rates are suspected to have varied widely in time or between communalects.

2. When the retention rates of the individual items of the sample wordlist are unequal, in which case the basic formula for the calculation of time depths no longer holds true.

If a sample wordlist consists of n items, with retention rates $r_1, r_2, r_3, \ldots r_n$ the most probable proportion of shared cognates (call it c) between two languages which have split t time units ago is given by the formula

$$c = \frac{1}{n}\left(r_1^{2t} + r_2^{2t} + r_3^{2t} + \ldots r_n^{2t}\right) \qquad (1)$$

In the very special case where all items have the same retention rate r (i.e. the same chance of surviving, unreplaced, over the unit period of time), the above formula reduces to

$$c = \frac{1}{n}\, nr^{2t} = r^{2t} \qquad (2)$$

on which traditional lexicostatistics is based. In a case where retention rates would be equally distributed over a range extending from a minimum retention r to a maximum retention R, formula (1) becomes approximately (the longer the wordlist, the closer the approximation)

$$c = \frac{R^{2t+1} - r^{2t+1}}{2t + 1} \qquad (3)$$

a far cry from formula (2). Other distributions of the retention rate of the different items of a standard wordlist would result in different formulas again, and it is almost certain that Hattori's proposal to replace formula (2) by a formula of the form

$$c = r^{xt} \qquad (4)$$

where x is a function of t and of the number of languages under scrutiny, stems from a purely empirical attempt at alleviating the effects of grossly unequal retention rates in his data.

Dyen, James, and Cole (1967) have given a sound mathematical method for palliating such effects. Their article, however, is quite beyond the reach of those who would most benefit from it, and the method involves such lengthy, repeated computations that it is doubtful whether it can be put to much practical use; neither can it allow for retention rates which vary in time or between communalects.

Not only are items with a very high or a very low retention rate devoid of any useful information, but their presence vitiates the very formula on which traditional lexicostatistics is based, and they should therefore be disregarded when calculating percentages of shared cognates. Disregarding sizable parts of painstakingly elicited wordlists may appear drastic, but there is no virtue in retaining garbage for the only reason that it was collected in the first place.

Granted that the above precautions have been taken, two more conditions should be met.

1. The wordlists, pruned of their useless items, should be as long as possible. It seems, from the experiments conducted, that 200 items constitute a barest minimum.

2. Loanwords should be scored as missing items. Even though this principle runs against all past practice, it remains true that loanwords are extraneous to the genetic model of inherited lexicon and that scoring them as non-cognates is as wrong as scoring them as cognates, since it amounts to treating noise as information.

SOME ADVICE

One should avoid the maximum-percentage method, as it gives sizably less reliable results. Where the lack of computer facilities or of a

programmable pocket calculator makes the use of the mean-percentage method impractical, the minimum-percentage method seems to offer a close second best; but time depths calculated on the base of minimal percentages are likely to be grossly overestimated.

If confidence levels are to be used at all they should be chosen fairly low, definitely way below the figure of 0.95 so often advocated, although I would personally favour the use of a zero tolerance or of a preset tolerance of a few percentage points, which in most cases will be so close to tolerances computed from sensible confidence levels that it will make no practical difference, these methods being on the whole rather unreliable.

Some authors, on the grounds that cognate percentages are already subject to error anyway, suggest that they should be rounded off to the nearest 5 or 10 percentage points before further processing. Such a practice does nothing but introduce more errors into already error-tainted data.

CORRELATION-BASED METHODS

The n-way splitting algorithm applied to unrecomputed linear-correlation coefficients has so far been found to give the most reliable reconstructions of the early history of a language family.

As in the case of traditional lexicostatistical methods, loanwords should have been scored as missing items, as many as possible of those items which show extreme retention rates (such as 'two', 'five', 'eye', 'dead', etc. in the case of Austronesian languages) should have been disregarded in calculating percentages of shared cognates, and the resulting wordlists should still be as long as possible (at the very least 100 items). There is no need however for retention rates to have remained constant in time and equal across communalects.

METHODS BASED ON COGNATE DISTRIBUTION PATTERNS

REDUCED MUTATION ALGORITHM

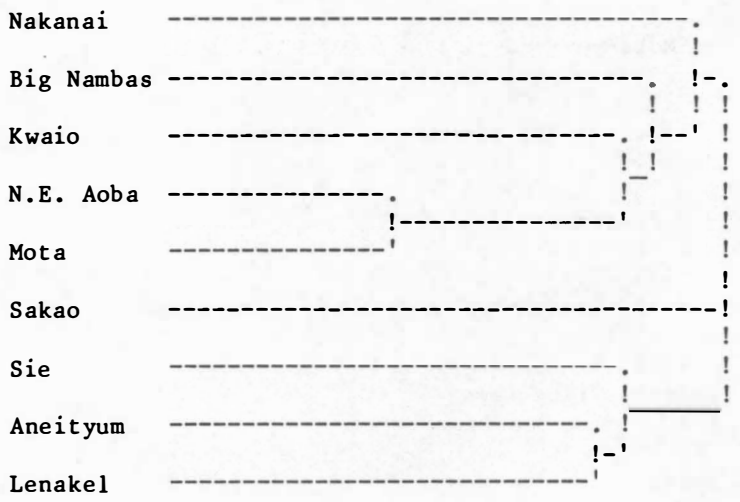
The reduced mutation algorithm has proved worthless and should be avoided. No listing of the corresponding program is therefore given in appendix II.

WILD-CARD ALGORITHM

The wild-card algorithm is very probably the most promising of all the methods examined. Although it does call for further development and testing, a program listing is given in appendix. Prospective users are warned that reconstructions beyond three successive mergings must be viewed with extreme skepticism.

APPENDIX 1

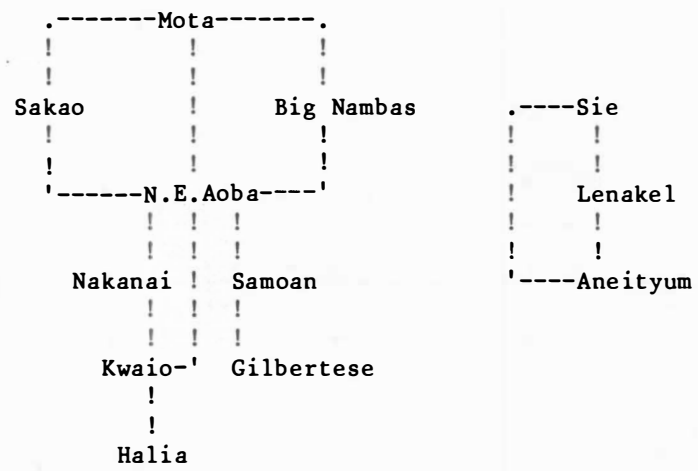
FIGURES and TABLES



[1]

Phylogenetic tree obtained by classic lexicostatistic methods (adapted from Tryon 1978:892)

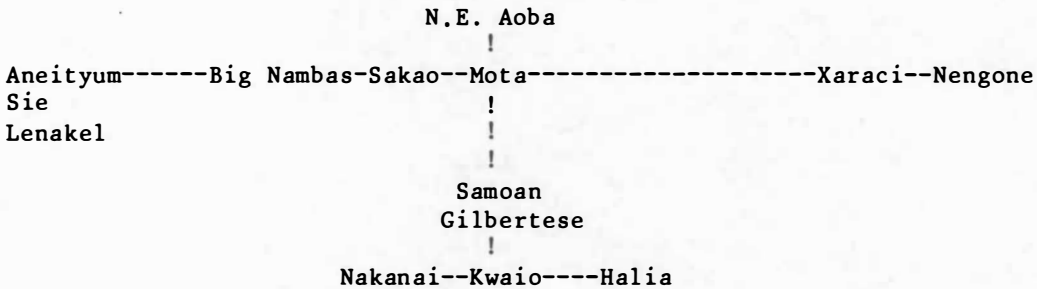
The lengths of the branches are proportional to time depth (absolute dates were not given). The graph is a subtree disconnected from a larger tree of 25 languages so that all New Hebridean languages should be represented. It contains two outsiders to the New Hebrides (Nakanai and Kwaio) and does not show the closer affinity of Sakao with Mota and N.E. Aoba than with Big Nambas.



[2]

Relationships between 12 Austronesian languages
(adapted from Tryon 1978:894)

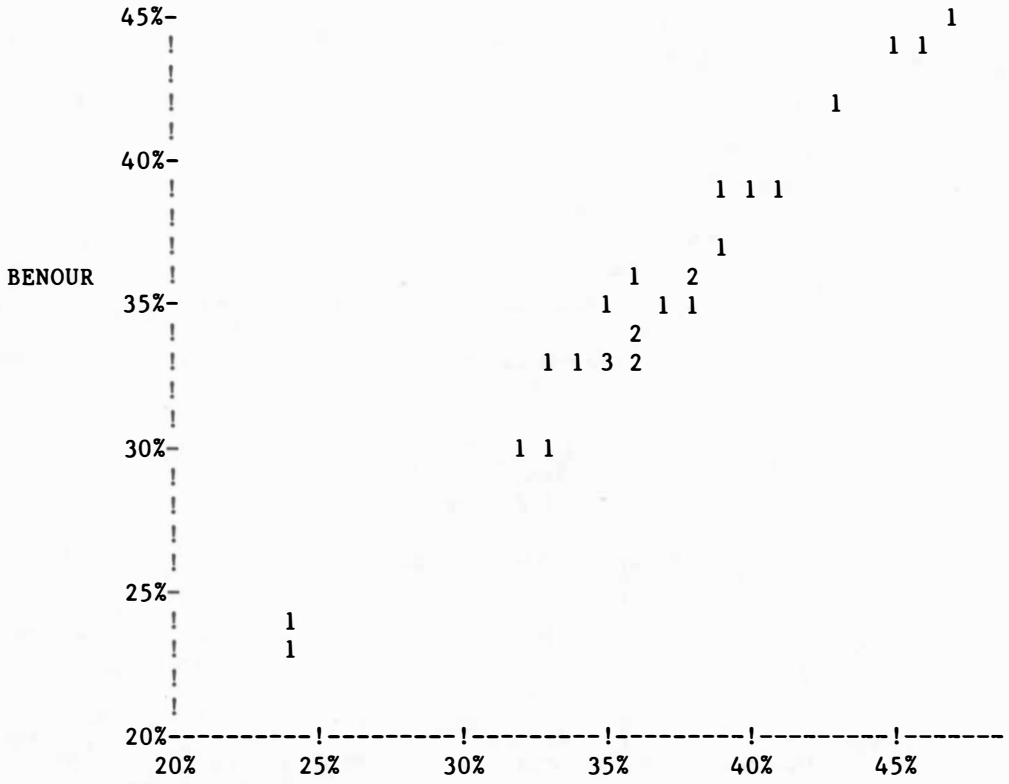
Languages not directly connected by a line have 25% or less common vocabulary. The closer relationship of Sakao with Mota and N.E. Aoba has become apparent, but the three southernmost New Hebridean languages (Sie, Lenakel, and Aneityum) now form a disconnected group.



[3]

Minimum-spanning tree obtained from linear correlation coefficients, using the single-linkage method (adapted from Tryon 1978:896)

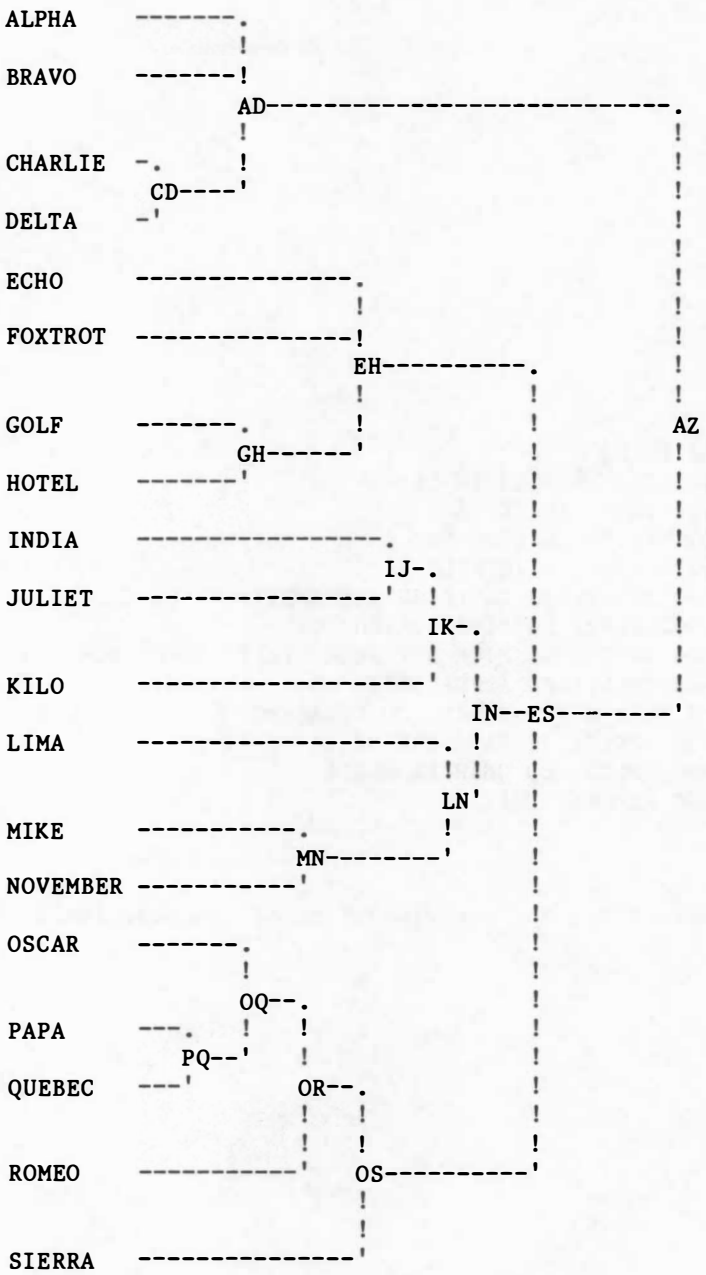
Branch length is roughly proportional to the inverse of the fifth power of the correlation between any two languages directly linked. The three southernmost New Hebridean languages now form a close group with correlations of 0.9 and above. The two earlier intruders, Nakanai and Kwaio, now appear distantly related via a chain Samoan-Gilbertese. The computations were carried out by hand on a Hewlett-Packard HP-25 and are somewhat marred by human error.



	LEMB'N	BENOUR		LEMB'N	BENOUR
TOGA	31.8	30.4	SAKAO	23.5	22.8
LEHALI	33.9	32.5	PORT SAND.	41.0	39.1
MOTLAV	35.0	32.5	LOPAXSIVIR	45.5	43.5
MOTA	35.0	32.5	AULUA	42.5	41.6
VATRATA	34.8	32.9	URIPIV	44.6	43.5
MOSINA	35.8	32.9	LABO	46.9	45.2
NUME	36.2	33.3	FONAH	39.1	36.9
KORO	35.8	33.9	MAAT	35.8	35.9
PETERARA	38.9	38.8	FILAKARA	33.1	32.5
WAILENGI	38.1	35.5	BURUMBA	35.2	34.8
NGWATUA	38.0	34.8	WORAVIU	36.9	34.9
RAGA	40.0	38.6	SESAKE	36.2	34.3
SA	38.2	35.7	FUTUNA	24.4	23.5
VALPEI	33.1	30.4			

[4]

Percentage scores of two very close dialects, Lembinwen and Benour (Malekula), one of which is represented by a shorter wordlist, show an extremely strong linear correlation.



[5]

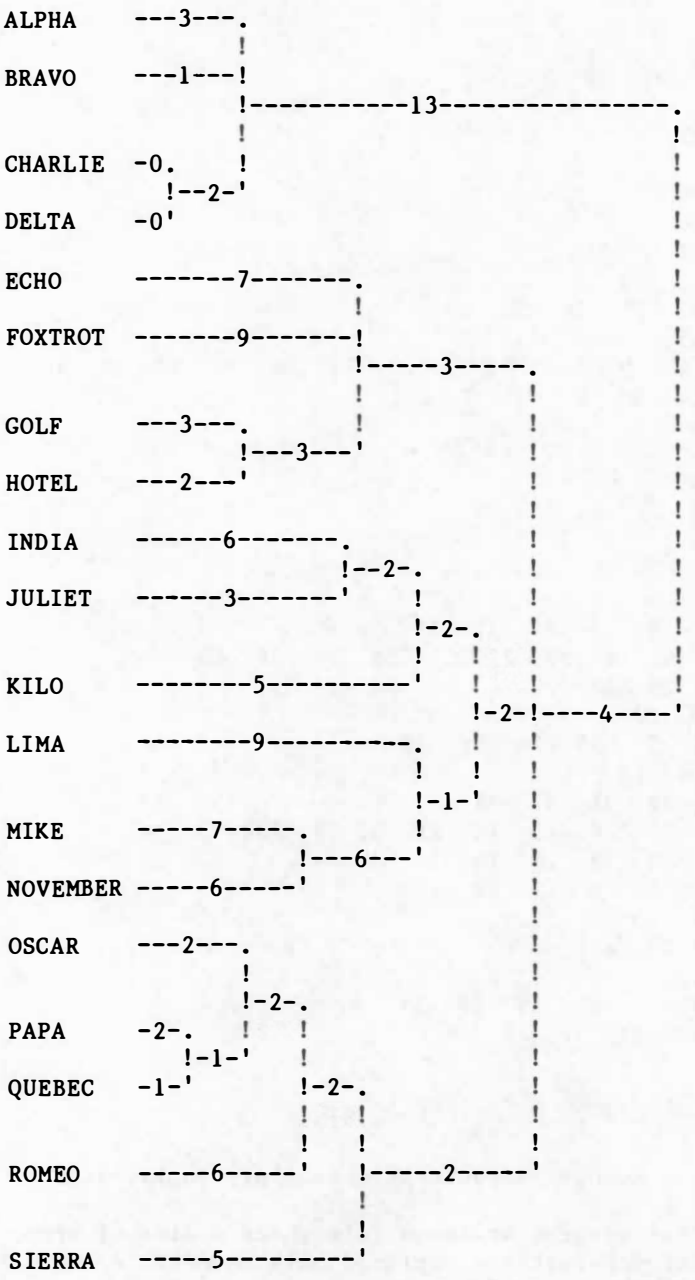
Language family to be simulated

The width of a character represents about 50 years.

```
CREATE: AZ
SPLIT: AZ AD ES
TIME: 500 SPLIT: ES EH IN OS
TIME: 700 SPLIT: IN IK LN
TIME: 800 SPLIT: LN LIMA MN
TIME: 850 SPLIT: IK IJ KILO
TIME: 1000 SPLIT: IJ INDIA JULIET SPLIT: OS OR SIERRA
TIME: 1100 SPLIT: EH ECHO FOXTROT GH
TIME: 1300 SPLIT: MN MIKE NOVEMBER SPLIT: OR OQ ROMEO
TIME: 1500 SPLIT: AD ALPHA BRAVO CD
SPLIT: GH GOLF HOTEL SPLIT: OQ OSCAR PQ
TIME: 1700 SPLIT: PQ PAPA QUEBEC
TIME: 1800 SPLIT: CD CHARLIE DELTA
TIME: 1900 REPORT: TEST
```

[6]

Instructions for the simulation of language family [5]



[7]

Language family effectively created by the simulation

The number of vocabulary items replaced between splits is shown on the branches. Sizable random deviations from the mean retention rate occurred, e.g. INDIA replaced twice as many items as JULIET over the same period of time.

```
ES   15   17   24   34
IN   22   25
LN   25
IK   24   38
IJ   20   36
OS   16   37
EH    8   12   13
MN   20   22   25   26   30   40
OR   31   35
AD    3    4    9   11   18   28   29   30   34   36   37   39   40
GH    5   14   35
OQ   33   34
PQ   30
CD    2   35
ALPHA    3    6   16
BRAVO    3
CHARLIE
DELTA
ECHO    2    8    9   24   31   37   39
FOXTROT    9   19   20   22   27   28   35   39   40
GOLF   10   29   32
HOTEL    3   17
INDIA    3   14   18   26   32   33
JULIET    7   13   16
KILO    1   28   31   37   38
LIMA    7    8    9   13   16   21   31   34   38
MIKE    3    7    8   11   16   19   32
NOVEMBER    5    6   11   12   29   35
OSCAR   14   25
PAPA   29   38
QUEBEC   12
ROMEO    9   11   23   25   28   36
SIERRA   15   21   22   29   34
```

[8]

Running account of vocabulary replacements

The simulation program writes a file where a list of items replaced is kept. QUEBEC for instance replaced only item#12 in the list inherited from its immediate ancestor.

ALPHA

BRAVO	925	BRAVO									
CHARLIE	875	925	CHARLIE								
DELTA	875	925	1000	DELTA							
ECHO	425	475	450	450	ECHO						
FOXTROT	350	400	375	375	650	FOXTROT					
GOLF	350	400	375	375	675	650	GOLF				
HOTEL	400	450	425	425	700	675	875	HOTEL			
INDIA	350	400	350	350	500	450	525	550	INDIA		
JULIET	400	425	375	375	600	550	575	600	775	JULIET	
KILO	425	475	425	425	650	550	575	600	675	750	KILO
LIMA	350	375	325	325	600	525	550	575	525	675	LIMA
MIKE	350	375	325	325	475	500	500	525	525	575	MIKE
NOVEMBER	375	400	375	375	500	525	575	575	500	550	NOVEMBER
OSCAR	425	450	425	425	625	525	625	650	575	625	OSCAR
PAPA	450	475	450	450	600	500	600	600	525	600	PAPA
QUEBEC	450	475	450	450	650	550	625	650	525	600	QUEBEC
ROMEO	450	475	450	450	600	525	550	575	500	600	ROMEO
SIERRA	500	525	475	475	625	550	625	625	550	650	SIERRA
	ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL	

KILO

LIMA	650	LIMA					
MIKE	525	525	MIKE				
NOVEMBER	550	475	700	NOVEMBER			
OSCAR	675	625	525	550	OSCAR		
PAPA	650	600	500	550	875	PAPA	
QUEBEC	650	600	525	575	900	925	QUEBEC
ROMEO	650	575	500	525	775	725	750	ROMEO
SIERRA	675	650	550	575	750	750	750	675	SIERRA
	KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM	

[9]

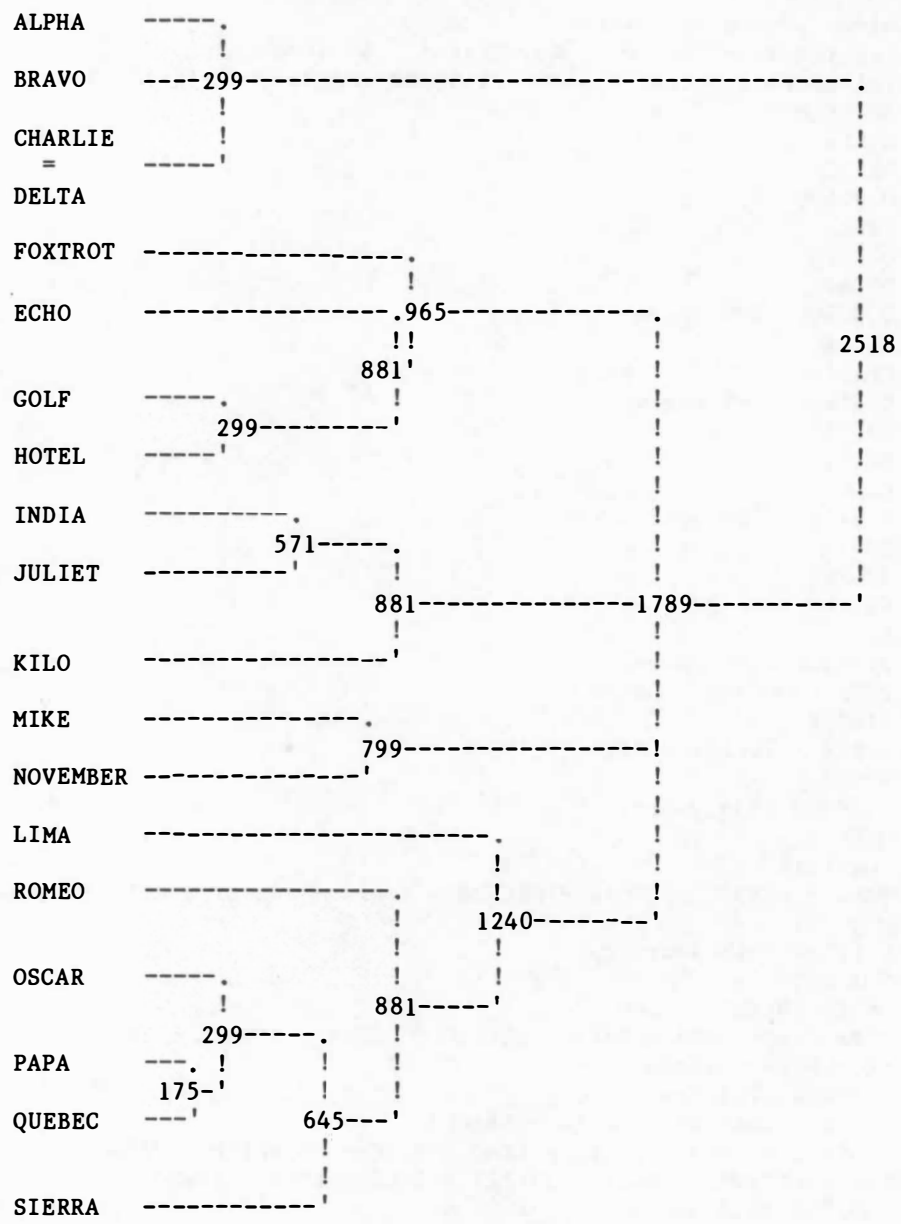
Cognate percentage table for languages ALPHA to SIERRA

Decimal points and leading zeroes are omitted, e.g. 55% appears as 550.

```
Minimum percentage method.
Retention rate: 0.8000. Tolerance: 0.0000
/ CHARLIE
/ DELTA
  1.00000    0 years.
/ ALPHA
/ BRAVO
/ CHARLIE - DELTA
  0.87500  299 years.
/ PAPA
/ QUEBEC
  0.92500  175 years.
/ GOLF
/ HOTEL
  0.87500  299 years.
/ OSCAR
/ PAPA - QUEBEC
  0.87500  299 years.
/ INDIA
/ JULIET
  0.77500  571 years.
/ OSCAR - PAPA - QUEBEC
/ SIERRA
  0.75000  645 years.
/ MIKE
/ NOVEMBER
  0.70000  799 years.
/ ECHO
/ GOLF - HOTEL
  0.67500  881 years.
/ INDIA - JULIET
/ KILO
  0.67500  881 years.
/ OSCAR - PAPA - QUEBEC - SIERRA
/ ROMEO
  0.67500  881 years.
/ ECHO - GOLF - HOTEL
/ FOXTROT
  0.65000  965 years.
/ LIMA
/ OSCAR - PAPA - QUEBEC - SIERRA - ROMEO
  0.57500 1240 years.
/ INDIA - JULIET - KILO
/ LIMA - OSCAR - PAPA - QUEBEC - SIERRA - ROMEO
/ ECHO - GOLF - HOTEL - FOXTROT
/ MIKE - NOVEMBER
  0.45000 1789 years.
/ ALPHA - BRAVO - CHARLIE - DELTA
/ INDIA - JULIET - KILO - LIMA - OSCAR - PAPA - QUEBEC - SIERRA -
ROMEO - ECHO - GOLF - HOTEL - FOXTROT - MIKE - NOVEMBER
  0.32500 2518 years.
```

[10a]

Output from traditional-lexicostatistics program,
minimum-percentage method, no tolerance



[10b]

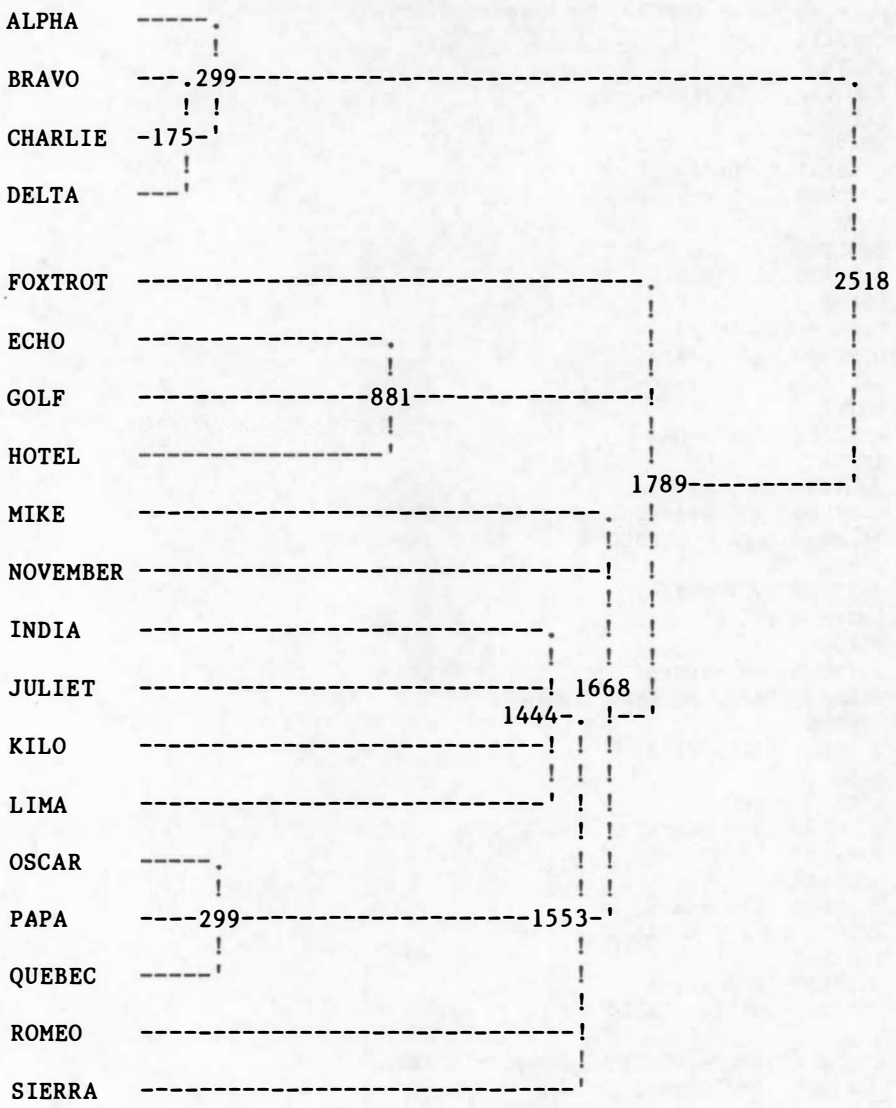
Tree drawn from computer output [10a],
minimum-percentage method, no tolerance

Numbers at the nodes give estimated time depths in years.

```
Minimum percentage method.
Retention rate: 0.8000.  Wordlists:   40 items.
Level of confidence: 0.95000 (1.96039 standard deviations).
/ CHARLIE
/ DELTA
/ BRAVO
  0.92500  175 years.
/ PAPA
/ QUEBEC
/ OSCAR
  0.87500  299 years.
/ ALPHA
/ CHARLIE - DELTA - BRAVO
  0.87500  299 years.
/ GOLF
/ HOTEL
/ ECHO
  0.67500  881 years.
/ INDIA
/ JULIET
/ KILO
/ LIMA
  0.52500 1444 years.
/ PAPA - QUEBEC - OSCAR
/ SIERRA
/ INDIA - JULIET - KILO - LIMA
/ ROMEO
  0.50000 1553 years.
/ MIKE
/ NOVEMBER
/ PAPA - QUEBEC - OSCAR - SIERRA - INDIA - JULIET - KILO - LIMA -
ROMEO
  0.47500 1668 years.
/ FOXTROT
/ GOLF - HOTEL - ECHO
/ MIKE - NOVEMBER - PAPA - QUEBEC - OSCAR - SIERRA - INDIA - JULIET -
KILO - LIMA - ROMEO
  0.45000 1789 years.
/ ALPHA - CHARLIE - DELTA - BRAVO
/ FOXTROT - GOLF - HOTEL - ECHO - MIKE - NOVEMBER - PAPA - QUEBEC -
OSCAR - SIERRA - INDIA - JULIET - KILO - LIMA - ROMEO
  0.32500 2518 years.
```

[11a]

Output from traditional lexicostatistics program,
minimum-percentage method,
tolerances computed for a 0.95 level of confidence



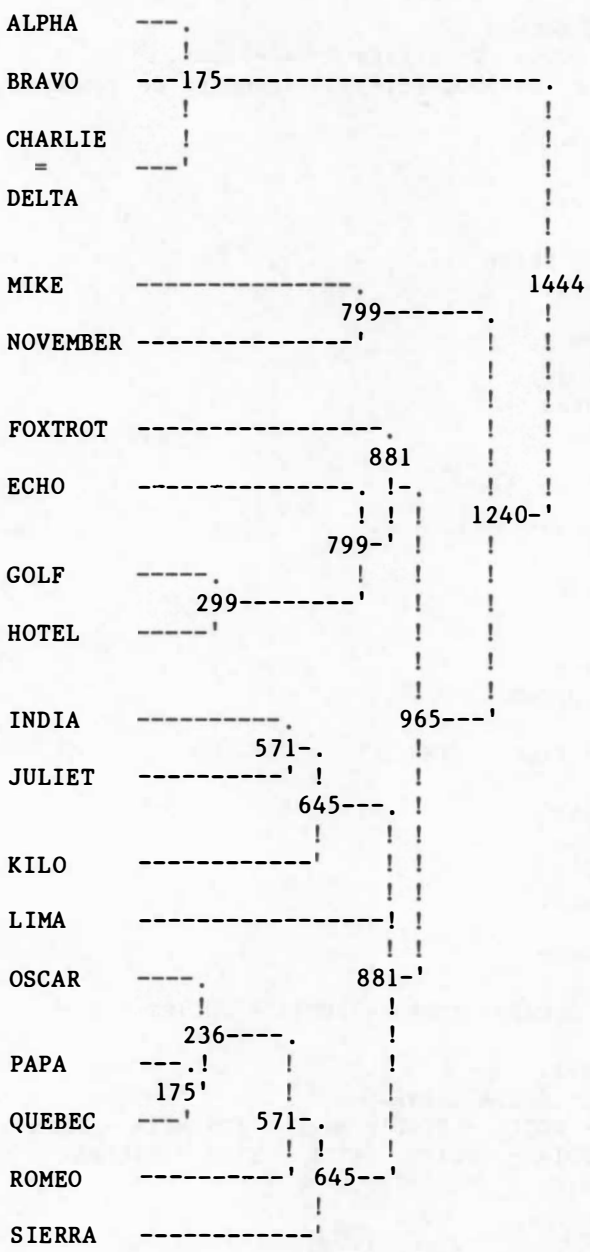
[11b]

Tree drawn from computer output [11a],
minimum-percentage method, 0.95 confidence level

```
Maximum percentage method.
Retention rate: 0.8000. Tolerance: 0.0000
/ CHARLIE
/ DELTA
  1.00000    0 years.
/ ALPHA
/ BRAVO
/ CHARLIE - DELTA
  0.92500  175 years.
/ PAPA
/ QUEBEC
  0.92500  175 years.
/ OSCAR
/ PAPA - QUEBEC
  0.90000  236 years.
/ GOLF
/ HOTEL
  0.87500  299 years.
/ INDIA
/ JULIET
  0.77500  571 years.
/ OSCAR - PAPA - QUEBEC
/ ROMEO
  0.77500  571 years.
/ INDIA - JULIET
/ KILO
  0.75000  645 years.
/ OSCAR - PAPA - QUEBEC - ROMEO
/ SIERRA
  0.75000  645 years.
/ ECHO
/ GOLF - HOTEL
  0.70000  799 years.
/ MIKE
/ NOVEMBER
  0.70000  799 years.
/ ECHO - GOLF - HOTEL
/ FOXTROT
  0.67500  881 years.
/ INDIA - JULIET - KILO
/ LIMA
/ OSCAR - PAPA - QUEBEC - ROMEO - SIERRA
  0.67500  881 years.
/ ECHO - GOLF - HOTEL - FOXTROT
/ INDIA - JULIET - KILO - LIMA - OSCAR - PAPA - QUEBEC - ROMEO -
SIERRA
  0.65000  965 years.
/ ECHO - GOLF - HOTEL - FOXTROT - INDIA - JULIET - KILO - LIMA - OSCAR
 - PAPA - QUEBEC - ROMEO - SIERRA
/ MIKE - NOVEMBER
  0.57500 1240 years.
/ ALPHA - BRAVO - CHARLIE - DELTA
/ ECHO - GOLF - HOTEL - FOXTROT - INDIA - JULIET - KILO - LIMA - OSCAR
 - PAPA - QUEBEC - ROMEO - SIERRA - MIKE - NOVEMBER
  0.52500 1444 years.
```

[12a]

Output from traditional-lexicostatistics program,
maximum percentage method, no tolerance.



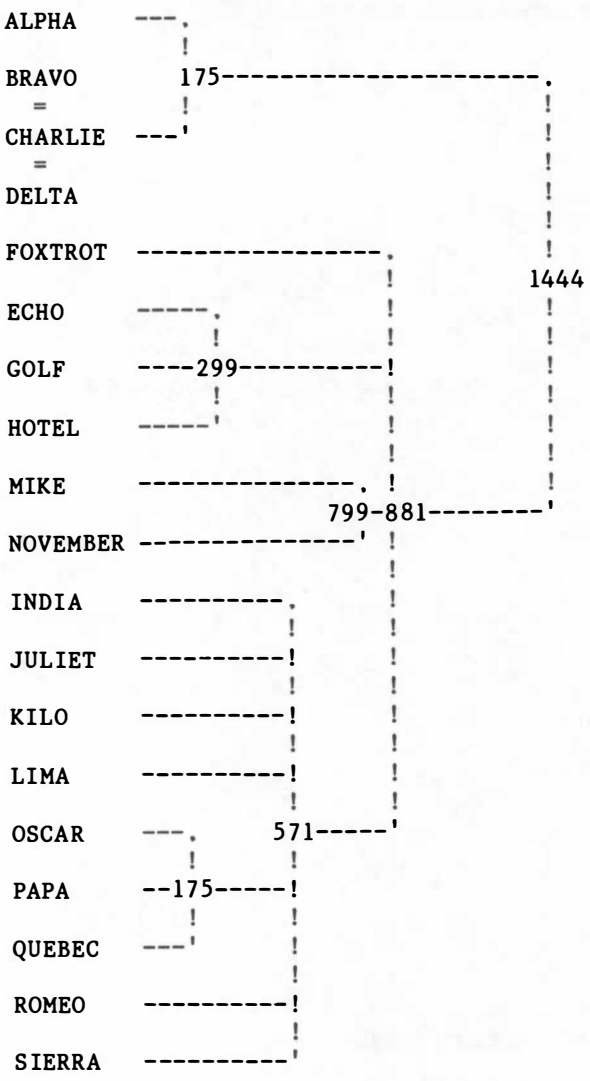
[12b]

Tree drawn from computer output [12a],
maximum percentage method, no tolerance

```
Maximum percentage method.
Retention rate: 0.8000.  Wordlists:   40 items.
Level of confidence: 0.95000 (1.96039 standard deviations).
/ CHARLIE
/ DELTA
/ BRAVO
  1.00000    0 years.
/ ALPHA
/ CHARLIE - DELTA - BRAVO
  0.92500  175 years.
/ PAPA
/ QUEBEC
/ OSCAR
  0.92500  175 years.
/ GOLF
/ HOTEL
/ ECHO
  0.87500  299 years.
/ INDIA
/ JULIET
/ KILO
/ LIMA
  0.77500  571 years.
/ PAPA - QUEBEC - OSCAR
/ ROMEO
/ INDIA - JULIET - KILO - LIMA
/ SIERRA
  0.77500  571 years.
/ MIKE
/ NOVEMBER
  0.70000  799 years.
/ FOXTROT
/ GOLF - HOTEL - ECHO
/ MIKE - NOVEMBER
/ PAPA - QUEBEC - OSCAR - ROMEO - INDIA - JULIET - KILO - LIMA -
SIERRA
  0.67500  881 years.
/ ALPHA - CHARLIE - DELTA - BRAVO
/ FOXTROT - GOLF - HOTEL - ECHO - MIKE - NOVEMBER - PAPA - QUEBEC -
OSCAR - ROMEO - INDIA - JULIET - KILO - LIMA - SIERRA
  0.52500 1444 years.
```

[13a]

Output from traditional-lexicostatistics program,
maximum percentage method,
tolerances computed for 0.95 confidence level



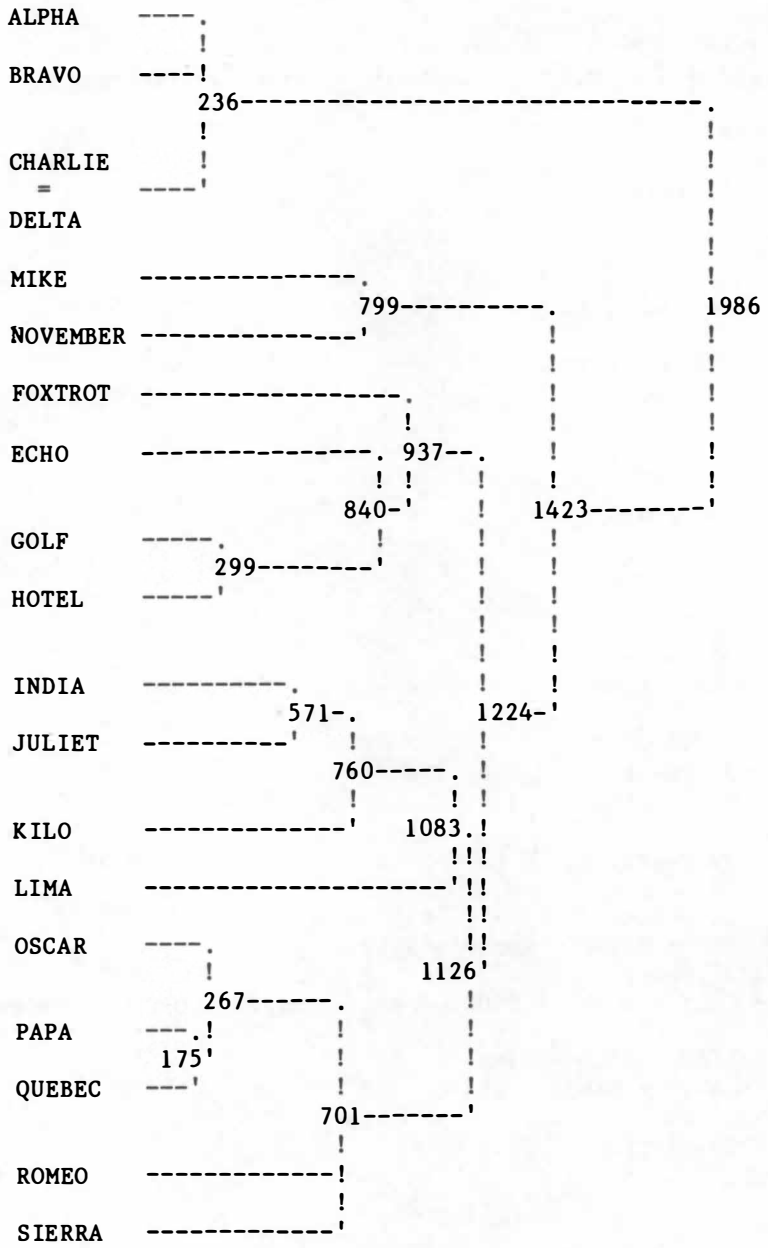
[13b]

Tree drawn from computer output [13a],
maximum-percentage method, 0.95 confidence level

```
Mean percentage method.
Retention rate: 0.8000. Tolerance: 0.0000
/ CHARLIE
/ DELTA
  1.00000    0 years.
/ ALPHA
/ BRAVO
/ CHARLIE - DELTA
  0.90500  224 years.
/ PAPA
/ QUEBEC
  0.92500  175 years.
/ OSCAR
/ PAPA - QUEBEC
  0.88750  267 years.
/ GOLF
/ HOTEL
  0.87500  299 years.
/ INDIA
/ JULIET
  0.77500  571 years.
/ OSCAR - PAPA - QUEBEC
/ ROMEO
/ SIERRA
  0.73929  677 years.
/ INDIA - JULIET
/ KILO
  0.71250  760 years.
/ MIKE
/ NOVEMBER
  0.70000  799 years.
/ ECHO
/ GOLF - HOTEL
  0.68750  840 years.
/ ECHO - GOLF - HOTEL
/ FOXTROT
  0.65833  937 years.
/ INDIA - JULIET - KILO
/ LIMA
  0.61667 1083 years.
/ INDIA - JULIET - KILO - LIMA
/ OSCAR - PAPA - QUEBEC - ROMEO - SIERRA
  0.60500 1126 years.
/ ECHO - GOLF - HOTEL - FOXTROT
/ INDIA - JULIET - KILO - LIMA - OSCAR - PAPA - QUEBEC - ROMEO -
SIERRA
  0.57917 1224 years.
/ ECHO - GOLF - HOTEL - FOXTROT - INDIA - JULIET - KILO - LIMA - OSCAR
 - PAPA - QUEBEC - ROMEO - SIERRA
/ MIKE - NOVEMBER
  0.52981 1423 years.
/ ALPHA - BRAVO - CHARLIE - DELTA
/ ECHO - GOLF - HOTEL - FOXTROT - INDIA - JULIET - KILO - LIMA - OSCAR
 - PAPA - QUEBEC - ROMEO - SIERRA - MIKE - NOVEMBER
  0.41208 1986 years.
```

[14a]

Output from traditional-lexicostatistics program,
mean percentage method, no tolerance



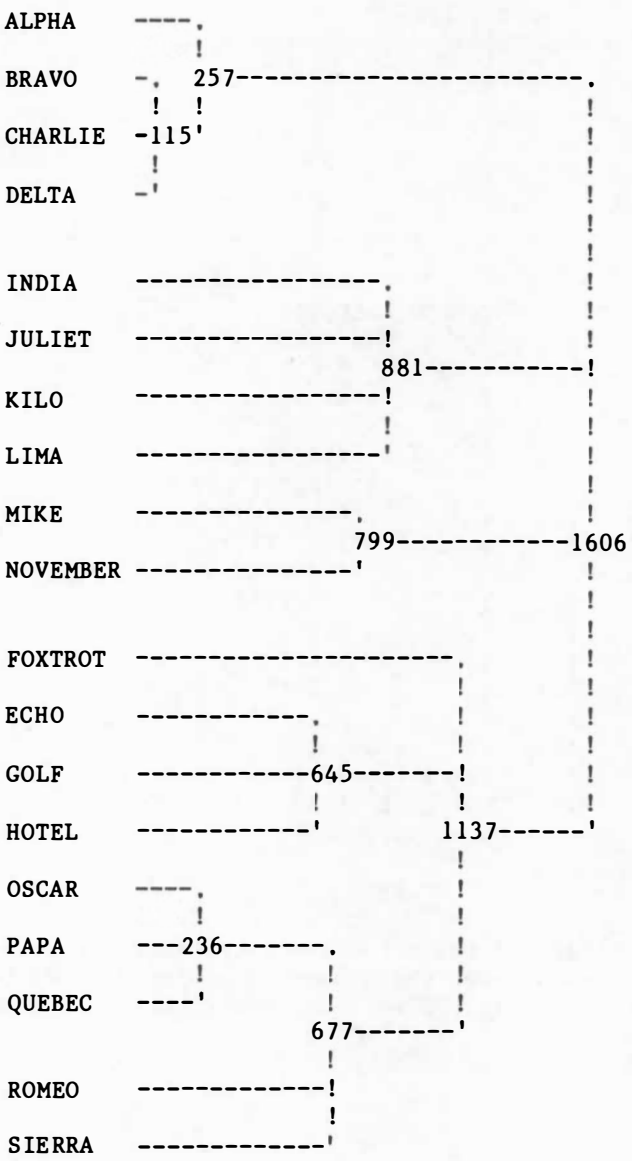
[14b]

Tree drawn from computer output [14b],
mean-percentage method, no tolerance

```
Mean percentage method.
Retention rate: 0.8000.  Wordlists:   40 items.
Level of confidence: 0.95000 (1.96039 standard deviations).
/ CHARLIE
/ DELTA
/ BRAVO
  0.95000  115 years.
/ PAPA
/ QUEBEC
/ OSCAR
  0.90000  236 years.
/ ALPHA
/ CHARLIE - DELTA - BRAVO
  0.89167  257 years.
/ GOLF
/ HOTEL
/ ECHO
  0.75000  645 years.
/ INDIA
/ JULIET
/ KILO
/ LIMA
  0.67500  881 years.
/ PAPA - QUEBEC - OSCAR
/ ROMEO
/ SIERRA
  0.73929  677 years.
/ MIKE
/ NOVEMBER
  0.70000  799 years.
/ FOXTROT
/ GOLF - HOTEL - ECHO
/ PAPA - QUEBEC - OSCAR - ROMEO - SIERRA
  0.60217 1137 years.
/ FOXTROT - GOLF - HOTEL - ECHO - PAPA - QUEBEC - OSCAR - ROMEO -
SIERRA
/ INDIA - JULIET - KILO - LIMA
/ ALPHA - CHARLIE - DELTA - BRAVO
/ MIKE - NOVEMBER
  0.48832 1606 years.
```

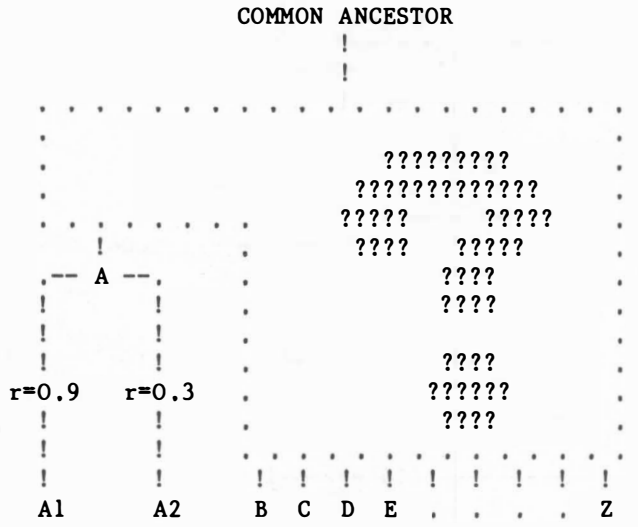
[15a]

Output from traditional-lexicostatistics program,
mean percentage method,
tolerances computed for 0.95 confidence level



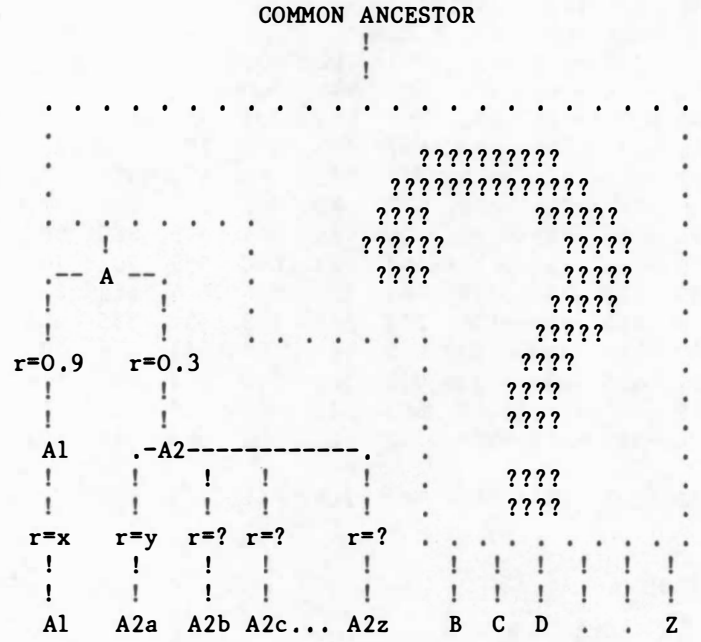
[15b]

Tree drawn from computer output [15a],
mean-percentage method, 0.95 confidence level



[16]

Whatever the branchings and the retention rates in the black box the ratio of related words over the number of compared words between A1 and any language B, C, D, ... Z will be around three times that of A2 with that language.



[17]

Whatever the branchings and the retention rates in the black box the ratio of related words over the number of compared words between A1 and any language B, C, D, ... Z will be around 3x/y times that of A2a with that language. But between A1 and any of the descendants A2b, ... A2z of A2 it will be a ratio of that of A2a with that same descendant, which ratio will vary depending on the retention rate of that descendant.

ALPHA

BRAVO	997	BRAVO									
CHARLIE	988	991	CHARLIE								
DELTA	988	991	1000	DELTA							
ECHO	-606	-638	-600	-600	ECHO						
FOXTROT	-722	-738	-701	-701	903	FOXTROT					
GOLF	-683	-695	-660	-660	883	946	GOLF				
HOTEL	-686	-700	-661	-661	867	930	993	HOTEL			
INDIA	-648	-693	-680	-680	637	637	555	505	INDIA		
JULIET	-792	-803	-806	-806	599	582	589	533	958	JULIET	
KILO	-699	-747	-728	-728	677	640	645	587	883	922	KILO
LIMA	-774	-799	-795	-795	806	766	744	695	880	861	LIMA
MIKE	-819	-856	-837	-837	548	633	646	559	703	706	MIKE
NOVEMBER	-783	-813	-798	-798	601	687	644	614	665	631	NOVEMBER
OSCAR	-488	-524	-506	-506	700	568	588	531	535	604	OSCAR
PAPA	-369	-412	-390	-390	650	496	508	502	498	507	PAPA
QUEBEC	-427	-465	-439	-439	639	504	570	512	498	514	QUEBEC
ROMEO	-419	-465	-440	-440	695	517	583	529	605	560	ROMEO
SIERRA	-538	-588	-539	-539	742	603	601	583	666	621	SIERRA

ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL

KILO

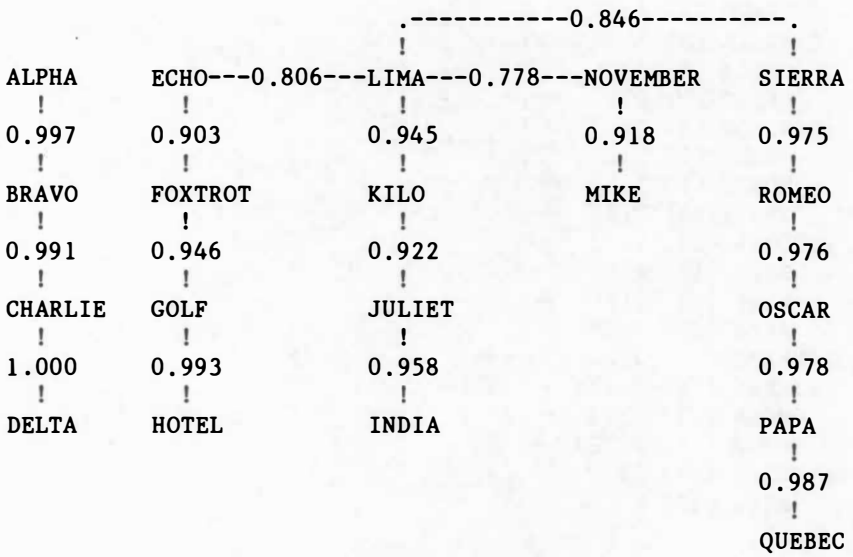
LIMA	945	LIMA						
MIKE	713	717	MIKE					
NOVEMBER	588	778	918	NOVEMBER				
OSCAR	750	783	524	581	OSCAR			
PAPA	678	710	472	491	978	PAPA		
QUEBEC	685	737	469	505	978	987	QUEBEC	
ROMEO	740	800	498	534	976	973	966	ROMEO
SIERRA	797	846	563	573	967	942	944	975

KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM	SIERRA

[18]

Linear correlation coefficients
computed from cognate percentages [9]

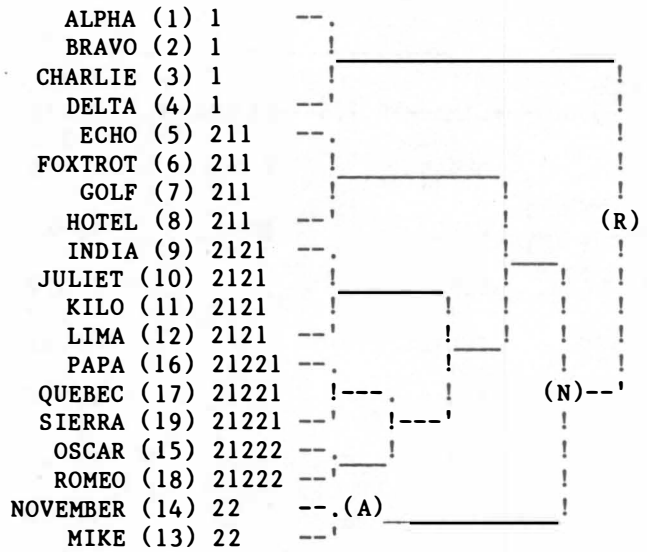
Decimal points and leading zeroes are omitted, e.g. -0.606 appears as -606.



[19]

Minimum-spanning tree, using the single-linkage method

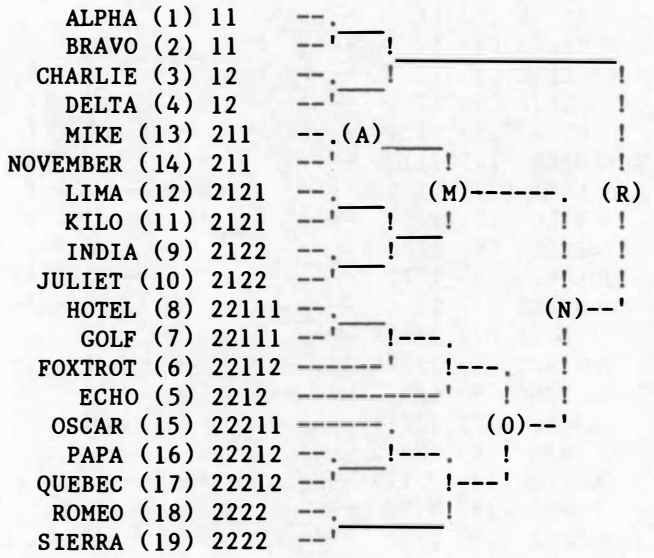
The ALPHA-DELTA chain is not positively correlated to any other. LIMA is the common link of four chains and its correlation coefficients with the three chain leaders ECHO, KILO, and SIERRA is not much lower than that of the weakest link in each chain.



[20]

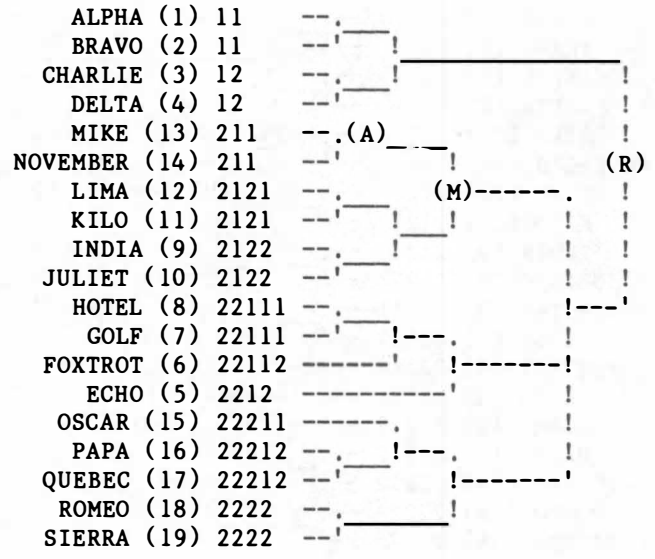
Phylogenetic tree reconstituted by the 2-way splitting algorithm, with correlations recomputed between successive splits

The tree was hand-drawn directly onto the computer output. The option of computing a new correlation table for each new group to be split makes it impossible to split groups of less than five languages.



[21]

Phylogenetic tree reconstituted by the 2-way splitting algorithm, correlations not recomputed between splits



[22]

Tree obtained by removing branch (N-O) from tree [21]

XARACI

NENGONE	814	NENGONE									
GILBERTESE	602	490	GILBERTESE								
SAMOAN	655	533	923	SAMOAN							
SAKAO	605	413	664	640	SAKAO						
B-NAMBAS	658	497	622	636	742	B-NAMBAS					
SIE	489	381	536	572	467	757	SIE				
LENAKEL	422	344	496	532	380	626	943	LENAKEL			
ANEITYUM	517	379	510	612	463	771	946	901	ANEITYUM		
MOTA	655	425	591	693	873	847	523	451	554	MOTA	
AOBA	518	298	644	591	884	761	493	349	457	866	AOBA
KWAIO	435	302	827	765	700	606	527	417	521	749	KWAIO
ROVIANA	-104	-235	420	276	462	223	123	37	132	369	ROVIANA
SENGGA	-169	-268	365	322	389	158	106	18	44	386	SENGGA
HALIA	119	-16	543	543	584	420	325	236	340	491	HALIA
NAKANAI	488	225	833	775	631	488	474	376	412	568	NAKANAI
YABEM	280	47	502	494	391	257	368	391	295	350	YABEM
GEDAGED	308	76	486	431	351	283	331	371	344	263	GEDAGED
KAIRIRU	327	145	517	531	362	272	347	376	372	273	KAIRIRU
ALI	322	222	546	514	304	326	399	455	294	309	ALI
TITAN	500	377	700	641	549	465	605	563	517	432	TITAN
MOTU	413	312	768	730	452	342	459	498	406	388	MOTU
KIRIWINA	280	153	569	528	309	320	490	468	531	243	KIRIWINA
TOBA BATAK	464	507	880	829	617	533	464	457	487	596	TOBA BATAK
MOR	408	80	556	625	373	375	542	520	537	460	MOR
	XAR	NEN	GIL	SAM	SAK	B-N	SIE	LEN	ANE	MOT	

AOBA

KWAIO	602	KWAIO									
ROVIANA	390	592	ROVIANA								
SENGGA	296	586	901	SENGGA							
HALIA	526	754	779	778	HALIA						
NAKANAI	500	860	554	472	718	NAKANAI					
YABEM	317	561	177	214	454	641	YABEM				
GEDAGED	225	442	46	-15	432	646	894	GEDAGED			
KAIRIRU	235	497	73	121	360	665	902	917	KAIRIRU		
ALI	190	466	42	2	411	650	840	908	896	ALI	
TITAN	399	600	143	121	429	724	768	789	875	877	TITAN
MOTU	364	623	145	193	575	756	751	741	788	835	MOTU
KIRIWINA	229	479	33	-26	395	645	764	816	868	823	KIRIWINA
TOBA BATAK	475	807	399	430	663	806	560	564	653	630	TOBA BATAK
MOR	201	558	195	265	397	689	733	705	739	753	MOR
	AOB	KWA	ROV	SEN	HAL	NAK	YAB	GED	KAI	ALI	

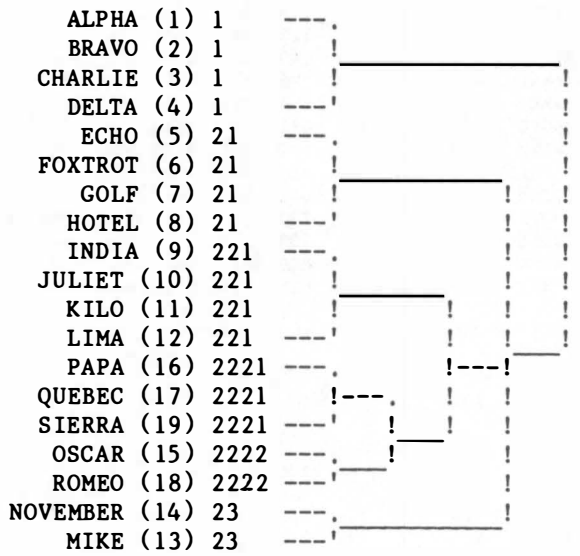
TITAN

MOTU	837	MOTU			
KIRIWINA	848	841	KIRIWINA		
TOBA BATAK	754	861	693	TOBA BATAK	
MOR	781	706	791	639	MOR
	TIT	MOT	KIR	TOB	

[23]

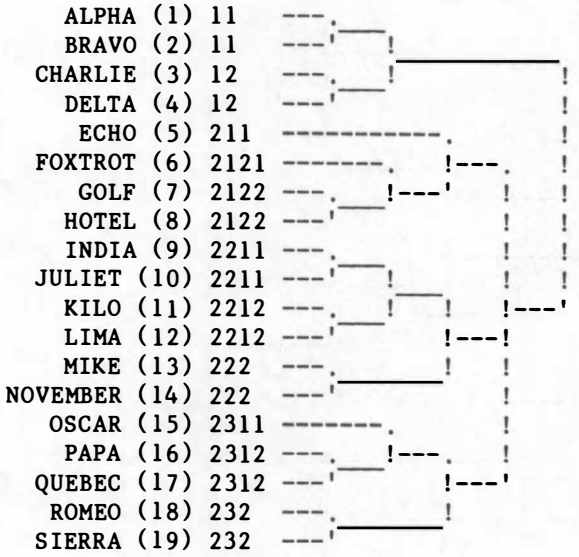
Correlations for 25 Austronesian languages
computed from Tryon's cognate percentages (1978)

Decimal points and leading zeroes are omitted. Apart from the clear dichotomy Xaraci-Nengone/Roviana-Sengga no obvious pattern emerges allowing any of the remaining languages to be grouped with Xaraci-Nengone, or Roviana-Sengga, or set up as one or more distinct groups.



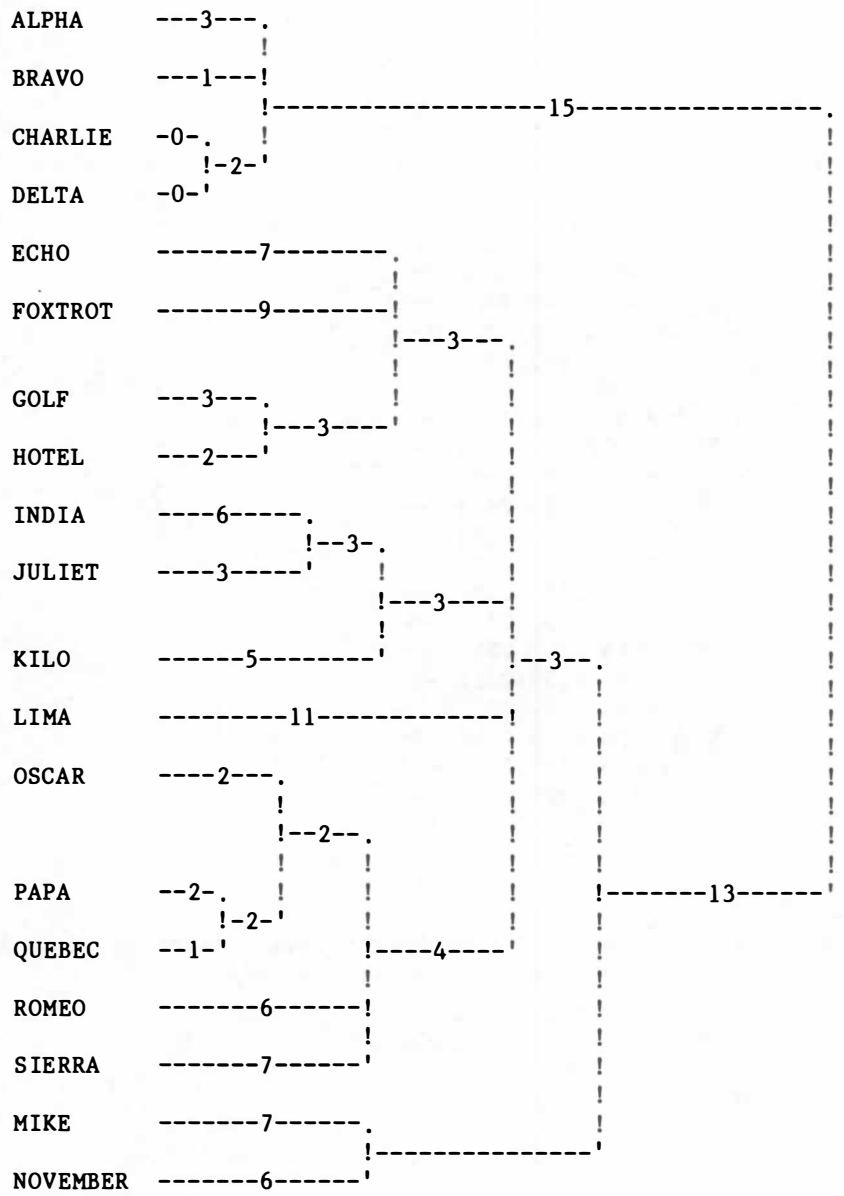
[24]

Phylogenetic tree reconstituted by the n-way splitting algorithm
(correlations recomputed between successive splits)



[25]

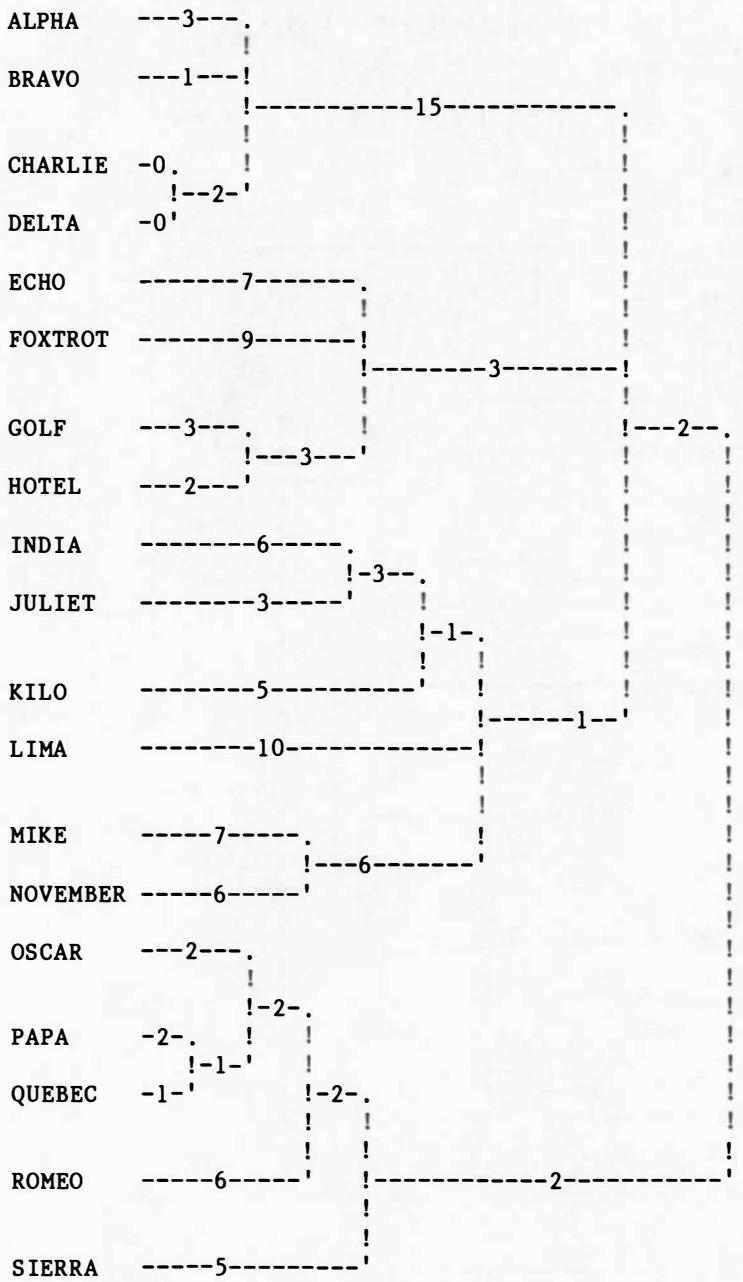
Phylogenetic tree reconstituted by the n-way splitting algorithm
(correlations not recomputed)



[26]

Phylogenetic tree reconstructed by the reduced mutation algorithm (Hartigan 1975:237-241)

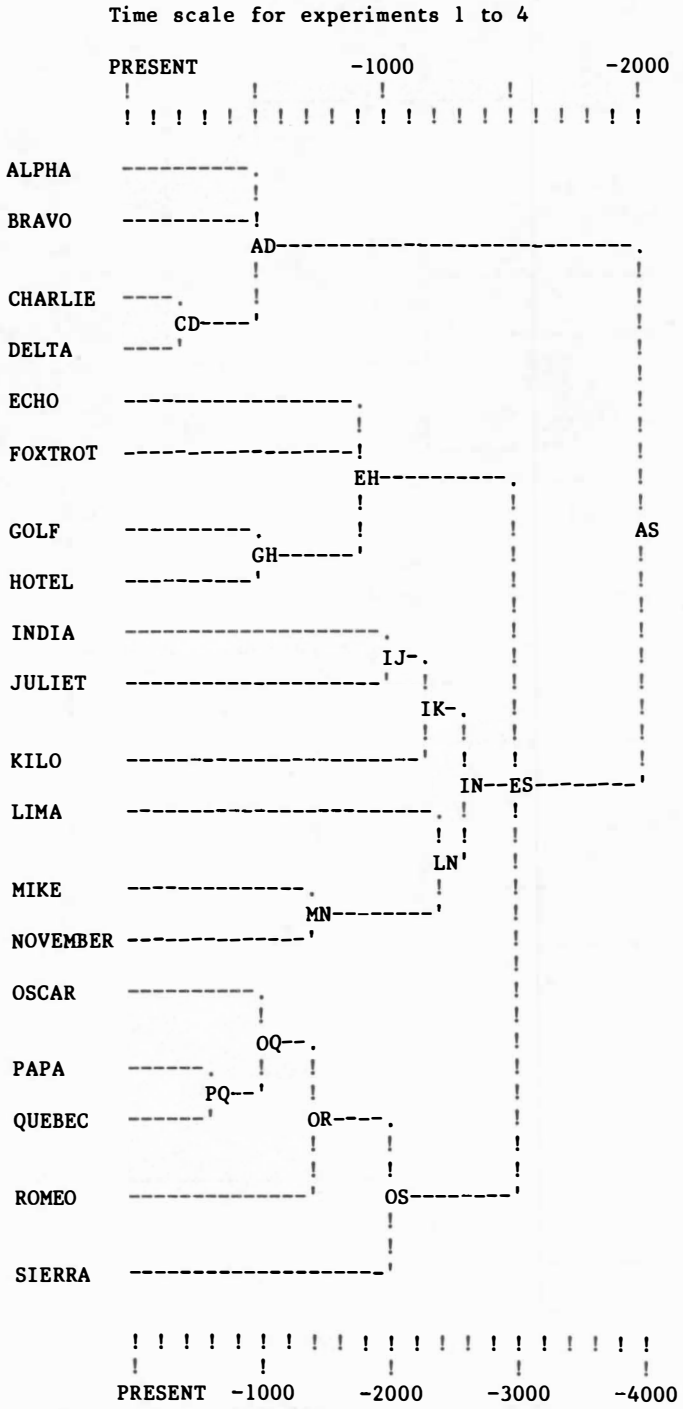
The numbers on the branches give the number of items replaced between successive splits.



[27]

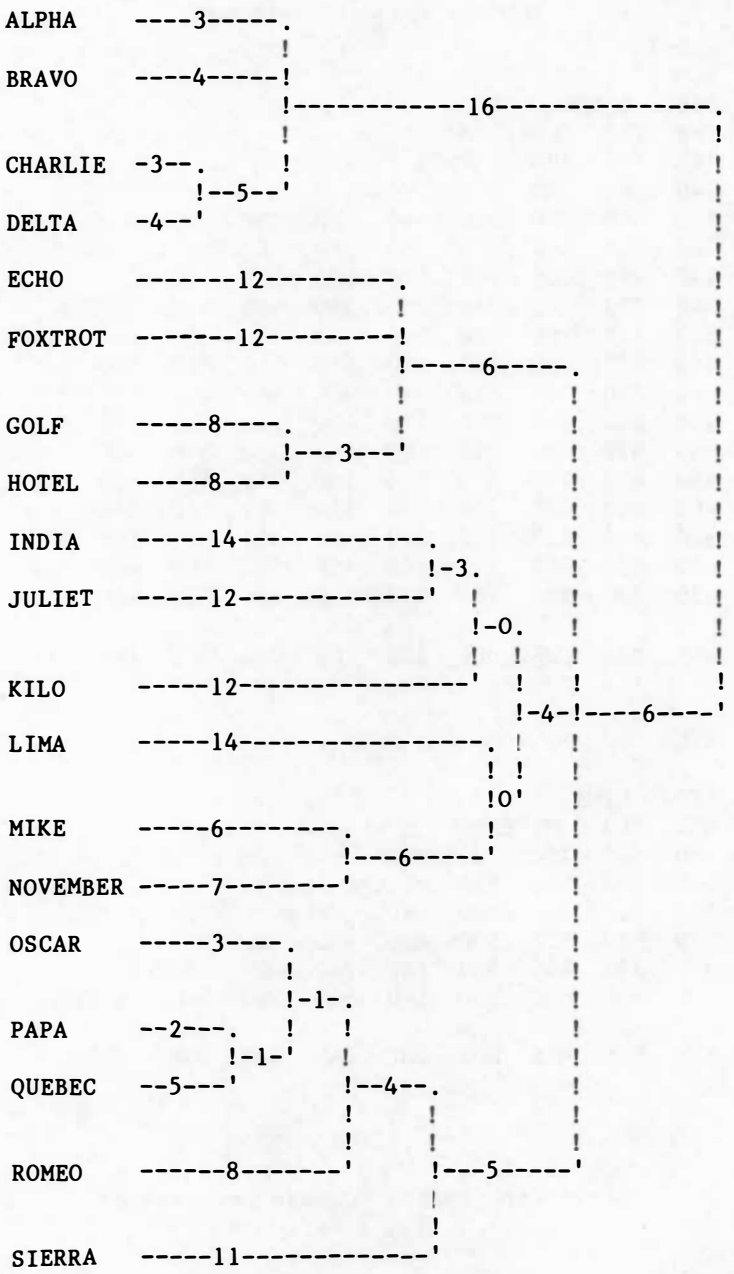
Phylogenetic tree reconstructed by the wild-card method

The number of items replaced between successive splits is shown on the branches. The predicted tree is remarkably similar to the actual tree near the leaves but becomes increasingly inaccurate near the root.



[000]

Basic tree of family to be simulated



[100]

First experiment: 50-item wordlists, 2000-year time depth

ALPHA

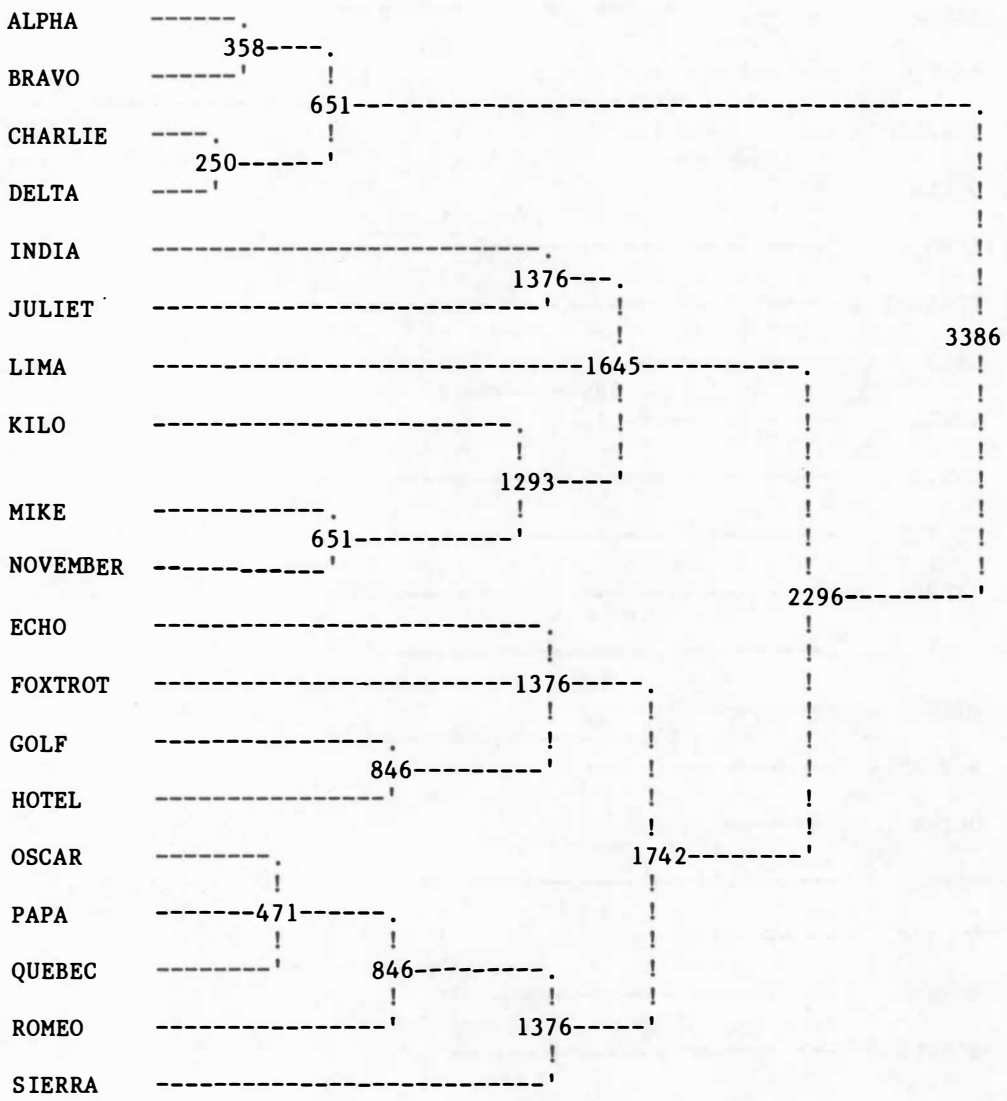
BRAVO	860	BRAVO									
CHARLIE	780	780	CHARLIE								
DELTA	760	760	900	DELTA							
ECHO	440	380	380	360	ECHO						
FOXTROT	380	320	340	320	560	FOXTROT					
GOLF	440	400	360	320	580	580	GOLF				
HOTEL	440	420	400	360	580	600	700	HOTEL			
INDIA	340	320	280	240	380	380	400	420	INDIA		
JULIET	360	340	300	280	460	420	480	480	560	JULIET	
KILO	460	400	420	380	480	480	480	460	520	540	KILO
LIMA	400	380	360	320	440	380	460	420	520	500	LIMA
MIKE	420	380	360	320	520	480	520	520	600	640	MIKE
NOVEMBER	460	420	400	360	480	460	520	560	540	560	NOVEMBER
OSCAR	460	420	400	360	560	580	560	580	500	500	OSCAR
PAPA	460	440	420	380	520	520	540	560	540	500	PAPA
QUEBEC	460	440	420	380	500	480	500	520	500	480	QUEBEC
ROMEO	440	420	380	340	480	480	500	500	440	440	ROMEO
SIERRA	420	380	380	360	520	500	480	480	460	420	SIERRA
	ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL	

KILO

LIMA	520	LIMA						
MIKE	600	580	MIKE					
NOVEMBER	580	560	760	NOVEMBER				
OSCAR	560	500	540	520	OSCAR			
PAPA	560	540	560	560	880	PAPA		
QUEBEC	520	520	520	500	820	880	QUEBEC	
ROMEO	540	500	480	500	800	760	700	ROMEO
SIERRA	480	500	520	520	640	660	640	560
	KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

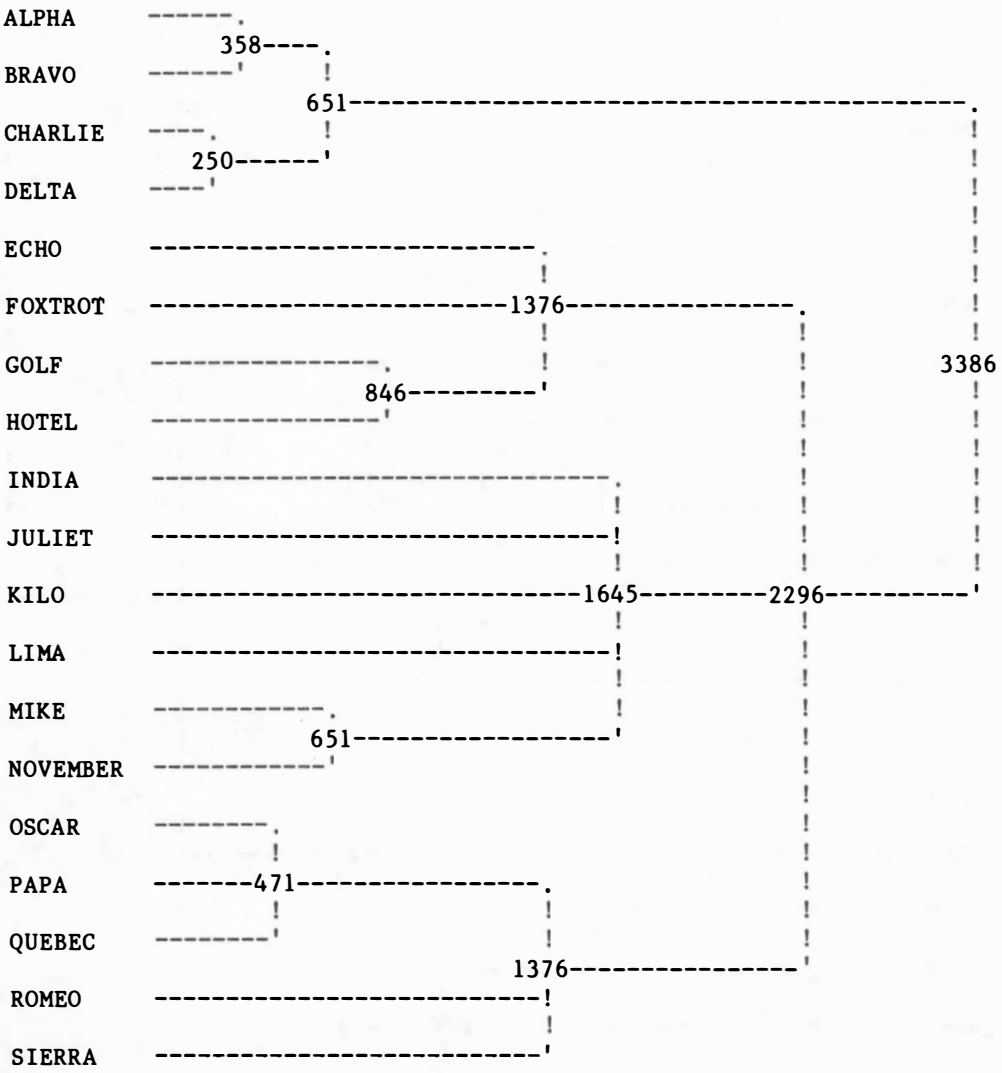
[110]

First experiment: cognate percentages



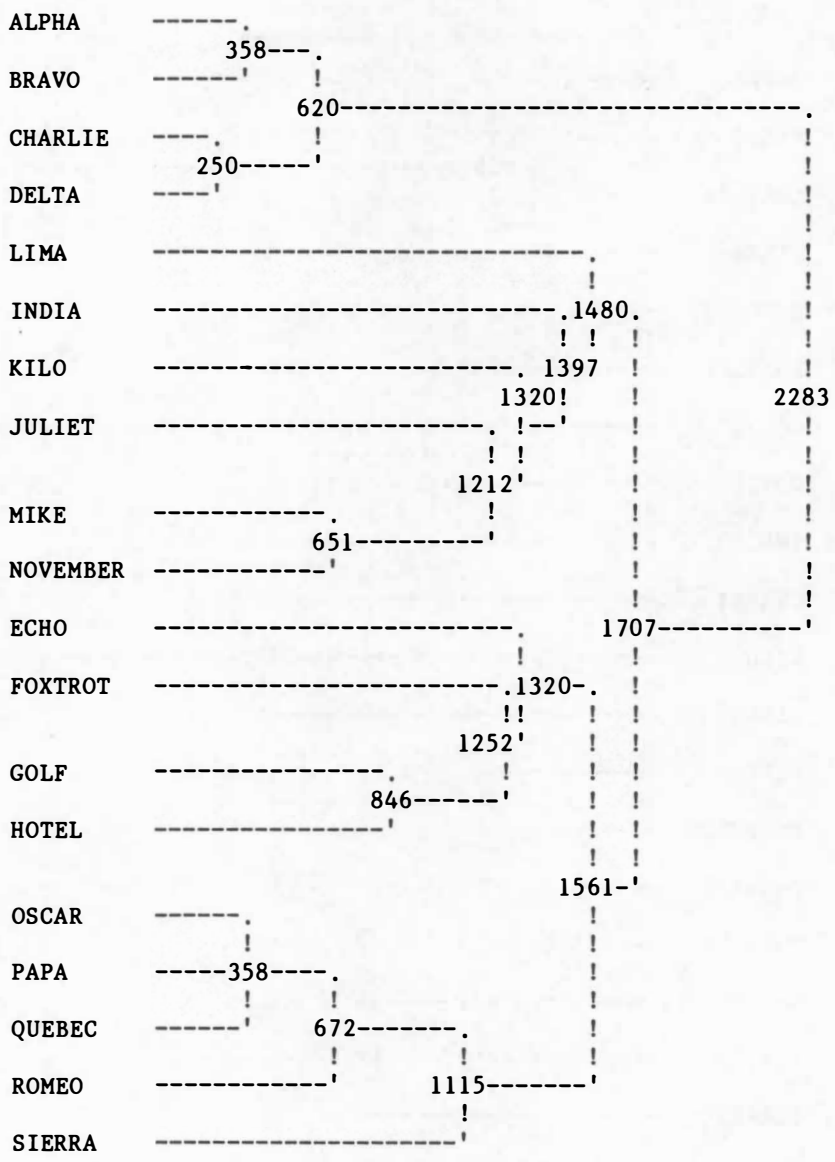
[121]

First experiment: traditional lexicostatistical techniques, minimum-percentage method, no tolerance



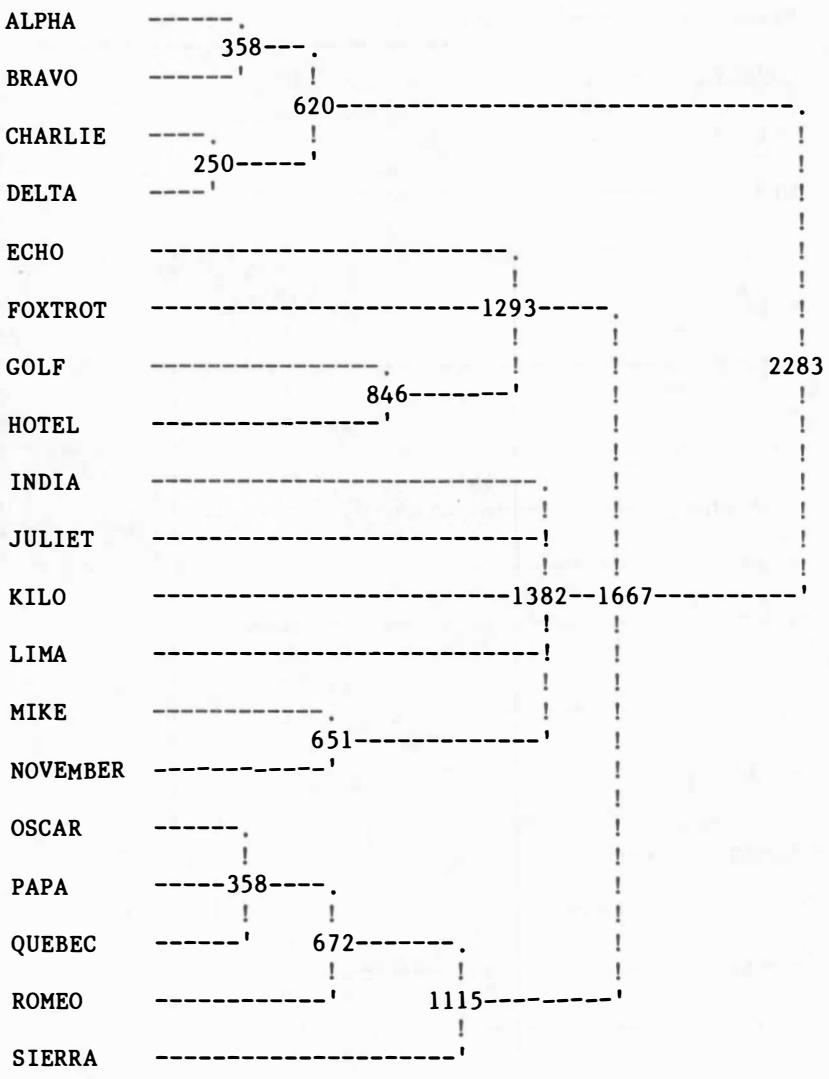
[122]

First experiment: traditional lexicostatistical techniques, minimum-percentage method, 0.5 confidence level



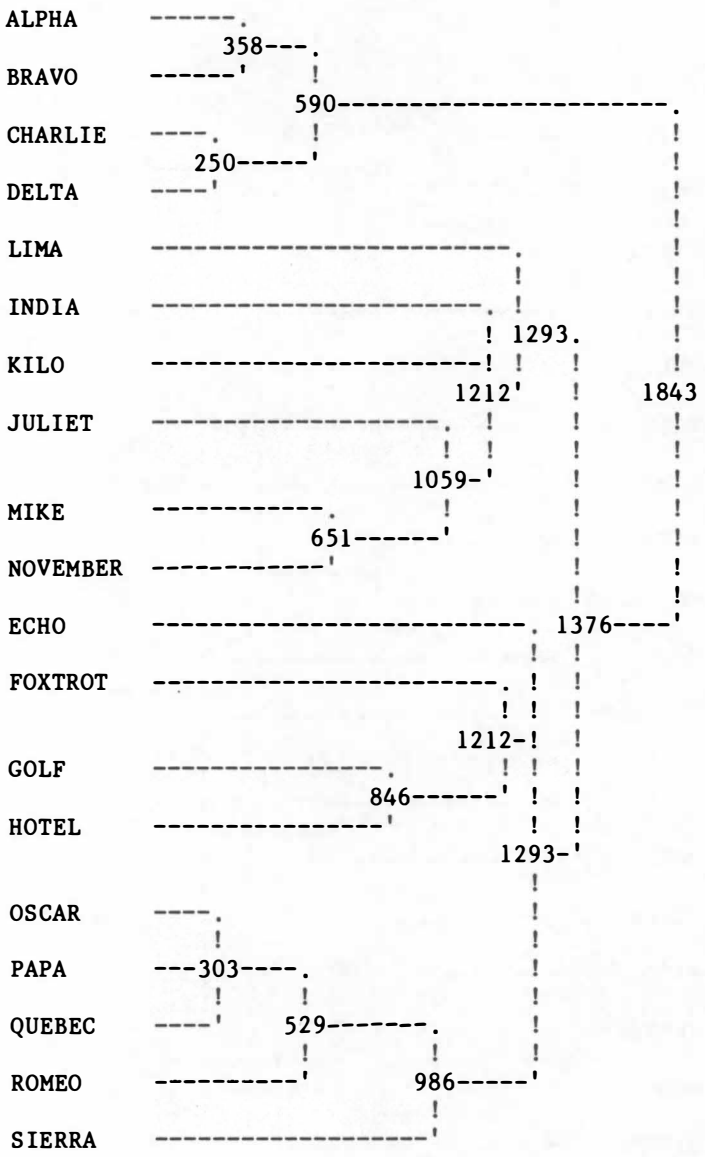
[123]

First experiment: traditional lexicostatistical techniques, mean-percentage method, no tolerance



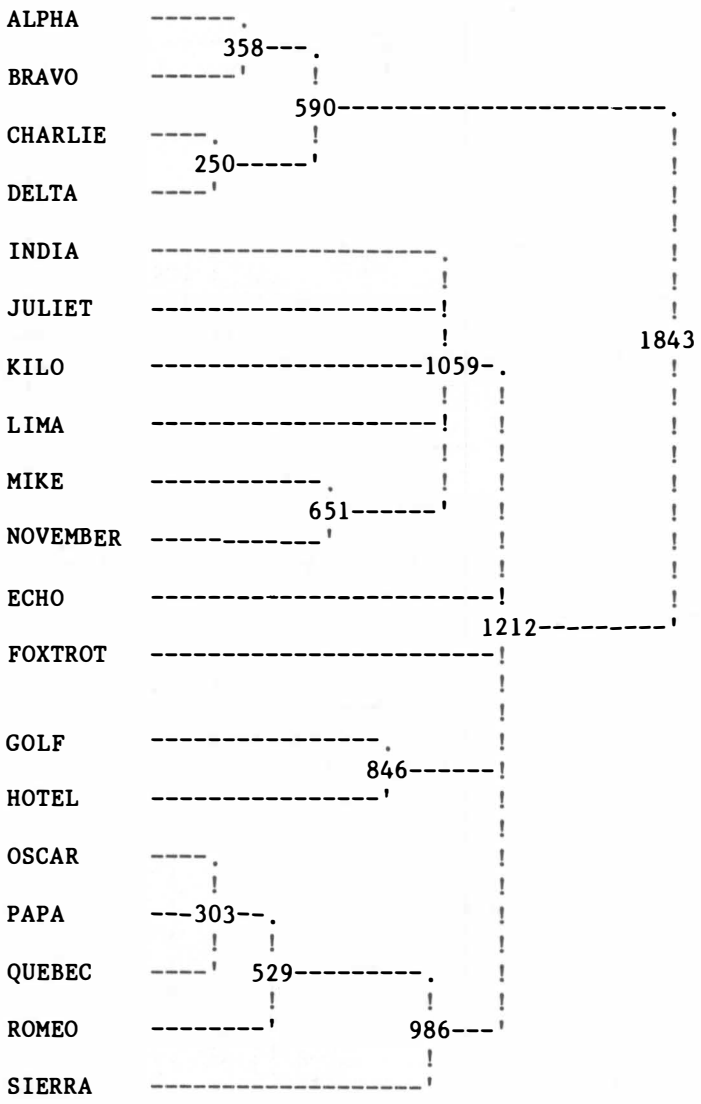
[124]

First experiment: traditional lexicostatistical techniques, mean-percentage method, 0.5 confidence level



[125]

First experiment: traditional lexicostatistical techniques, maximum-percentage method, no tolerance



[126]

First experiment: traditional lexicostatistical techniques, maximum-percentage method, 0.5 confidence level

ALPHA

BRAVO	993	BRAVO									
CHARLIE	962	961	CHARLIE								
DELTA	963	960	998	DELTA							
ECHO	-595	-530	-559	-535	ECHO						
FOXTROT	-590	-554	-585	-568	973	FOXTROT					
GOLF	-586	-555	-562	-522	932	921	GOLF				
HOTEL	-491	-502	-521	-471	904	897	958	HOTEL			
INDIA	-737	-702	-740	-728	490	438	455	339	INDIA		
JULIET	-770	-750	-782	-786	472	516	494	429	934	JULIET	
KILO	-724	-619	-711	-673	456	448	437	451	933	887	KILO
LIMA	-671	-633	-686	-663	379	461	327	362	952	891	LIMA
MIKE	-737	-701	-740	-720	357	399	426	382	892	933	MIKE
NOVEMBER	-641	-584	-624	-596	472	444	450	285	874	919	NOVEMBER
OSCAR	-469	-422	-451	-432	561	594	497	468	497	368	OSCAR
PAPA	-428	-404	-433	-416	520	563	415	389	508	404	PAPA
QUEBEC	-355	-326	-354	-330	494	559	413	386	500	346	QUEBEC
ROMEO	-373	-354	-362	-341	590	620	482	485	559	422	ROMEO
SIERRA	-538	-490	-523	-517	629	668	577	577	627	578	SIERRA

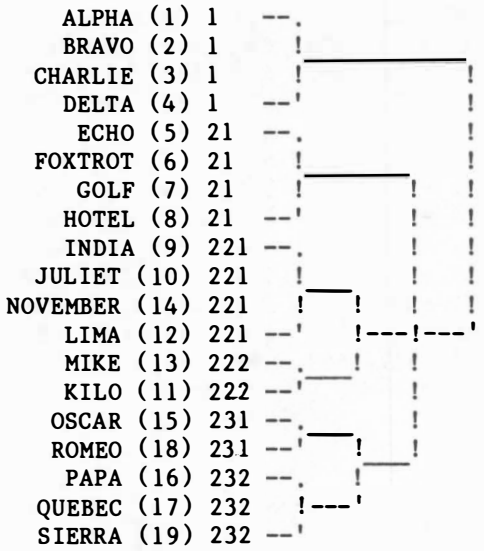
KILO

LIMA	930	LIMA						
MIKE	854	857	MIKE					
NOVEMBER	824	836	939	NOVEMBER				
OSCAR	535	559	287	306	OSCAR			
PAPA	565	569	304	269	963	PAPA		
QUEBEC	571	539	276	308	953	991	QUEBEC	
ROMEO	568	555	351	306	976	971	970	ROMEO
SIERRA	711	657	436	412	930	931	911	938

KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

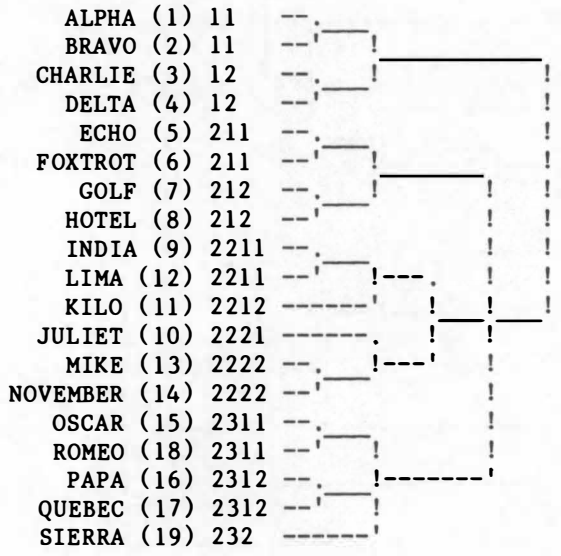
[130]

First experiment: correlation coefficients



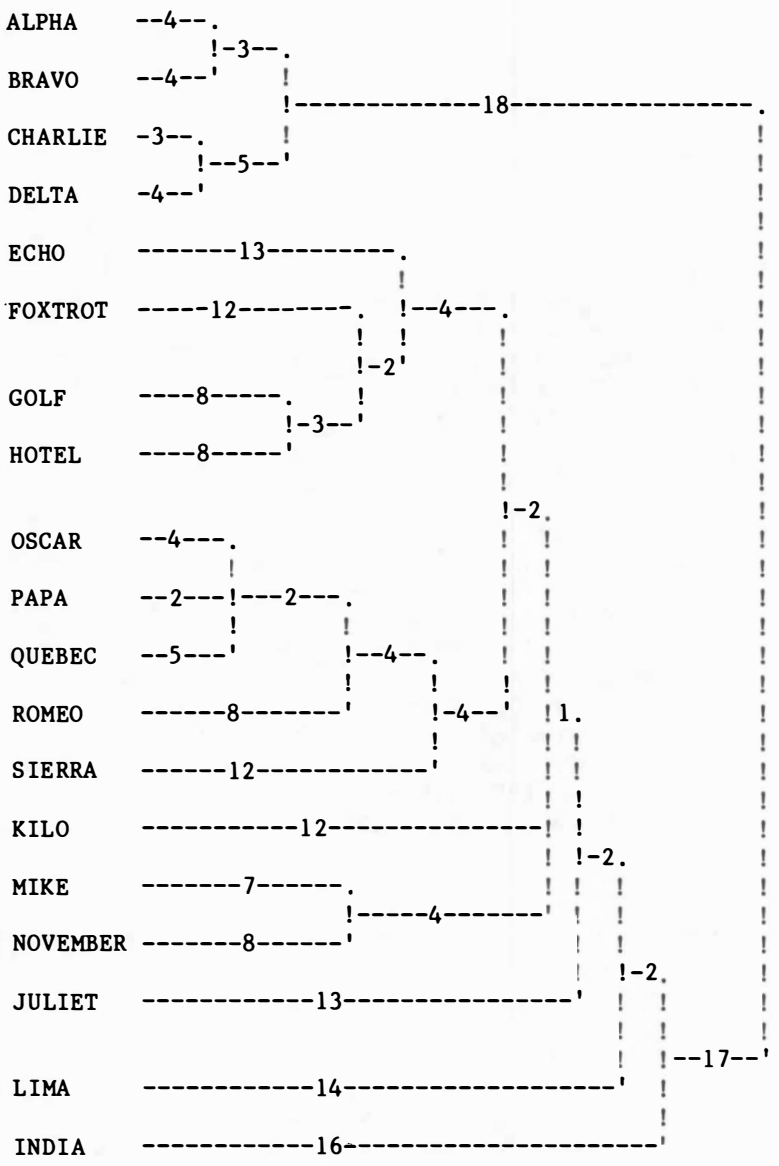
[147]

First experiment: n-way splitting algorithm, with recomputations



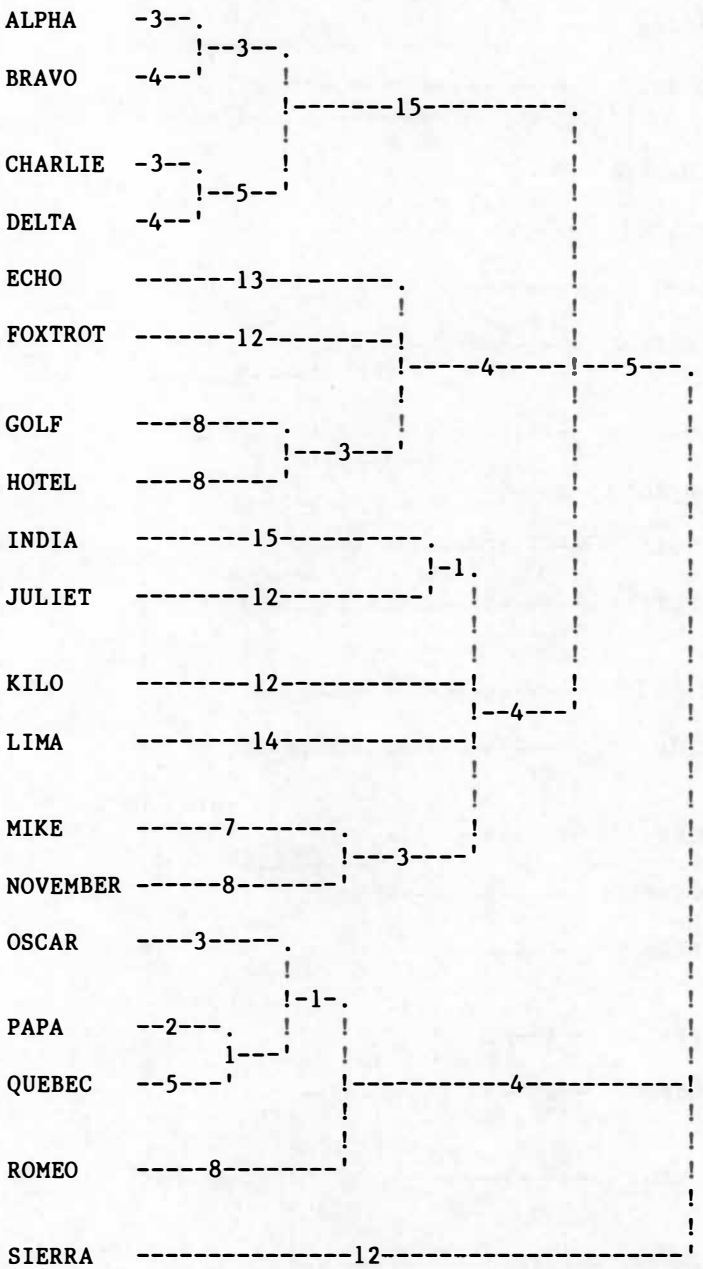
[148]

First experiment: n-way splitting algorithm, no recomputations



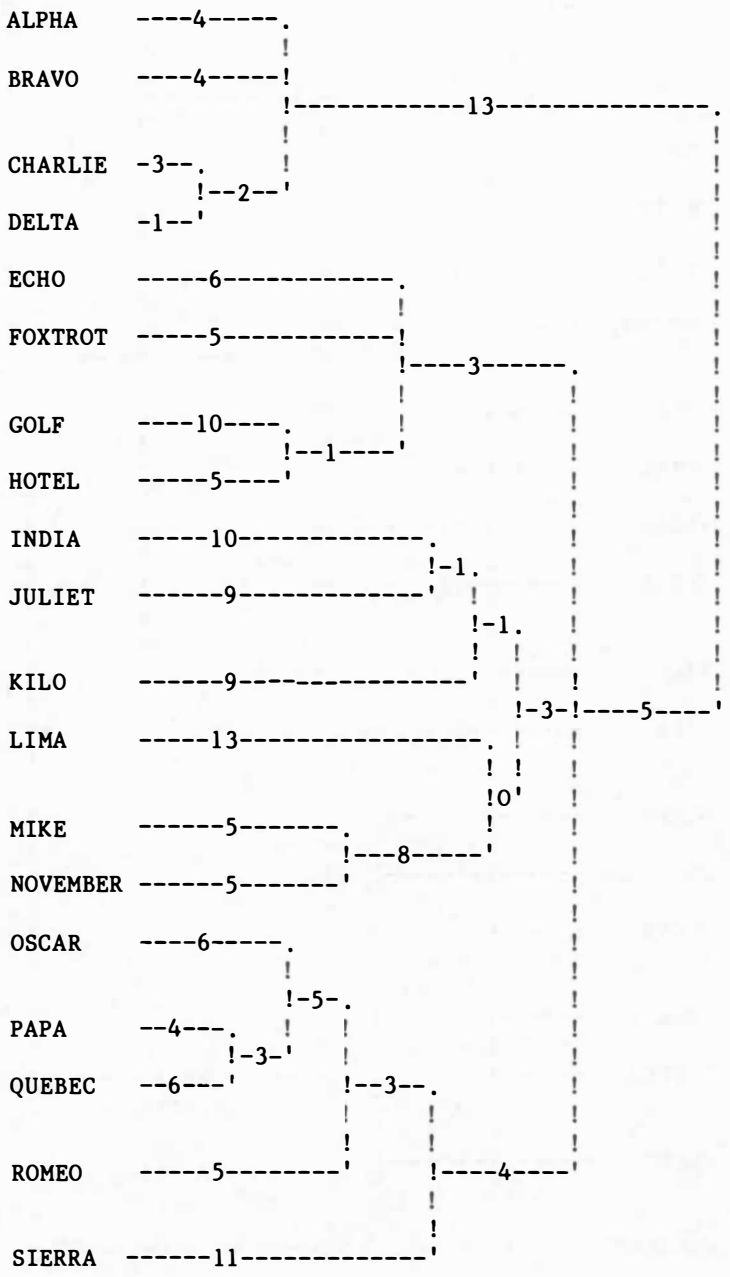
[150]

First experiment: reduced mutation algorithm



[160]

First experiment: wild-card algorithm



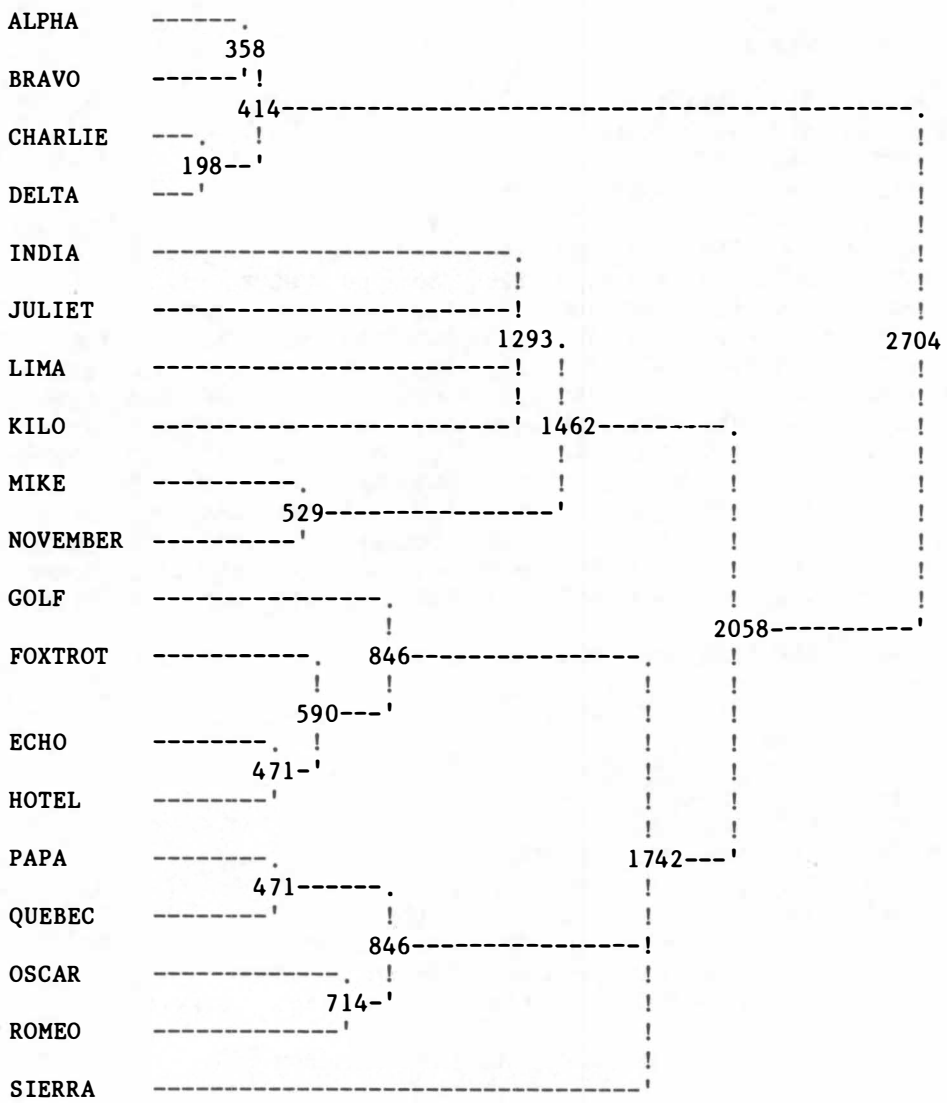
[200]

Second experiment: 50-item wordlists, 2000-year time depth

```
          ALPHA

   BRAVO  860   BRAVO
 CHARLIE  840  840   CHARLIE
   DELTA  860  860  920   DELTA
    ECHO  500  560  500  520   ECHO
 FOXTROT  500  540  500  520  780   FOXTROT
    GOLF  500  540  480  500  700  700   GOLF
   HOTEL  520  560  520  540  820  780  700   HOTEL
   INDIA  360  400  380  400  560  580  500  600   INDIA
  JULIET  400  420  400  420  580  600  480  620  660   JULIET
    KILO  460  500  480  500  680  700  560  720  640  620  KILO
    LIMA  360  380  360  380  580  560  520  620  580  660  LIMA
    MIKE  440  460  420  440  620  640  580  640  540  600  MIKE
NOVEMBER  420  460  420  440  580  580  520  600  540  600  NOVEMBER
   OSCAR  380  380  320  340  580  540  480  580  460  520  OSCAR
    PAPA  380  380  320  340  580  520  500  560  460  500  PAPA
  QUEBEC  420  420  360  380  540  500  480  540  420  480  QUEBEC
   ROMEO  460  440  400  420  640  640  540  660  560  580  ROMEO
  SIERRA  420  420  400  400  600  580  540  620  460  500  SIERRA

          ALP  BRA  CHA  DEL  ECH  FOX  GOL  HOT  IND  JUL


          KILO

    LIMA  600   LIMA
    MIKE  620  600   MIKE
NOVEMBER  560  580  800   NOVEMBER
   OSCAR  540  500  480  440   OSCAR
    PAPA  500  500  480  460  780   PAPA
  QUEBEC  460  440  460  440  720  820   QUEBEC
   ROMEO  580  540  540  500  740  720  700   ROMEO
  SIERRA  540  520  560  520  620  560  540  660   SIERRA

          KIL  LIM  MIK  NOV  OSC  PAP  QUE  ROM
```

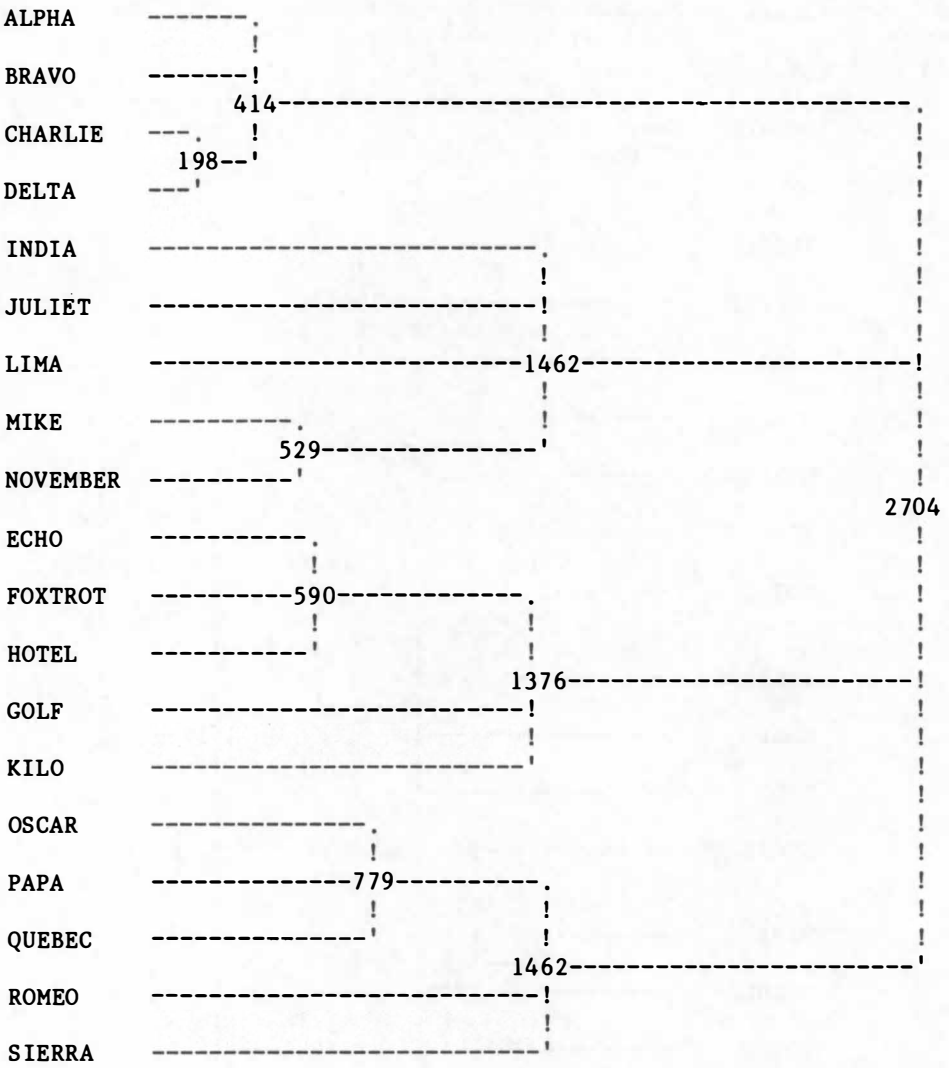
[210]

Second experiment: cognate percentages



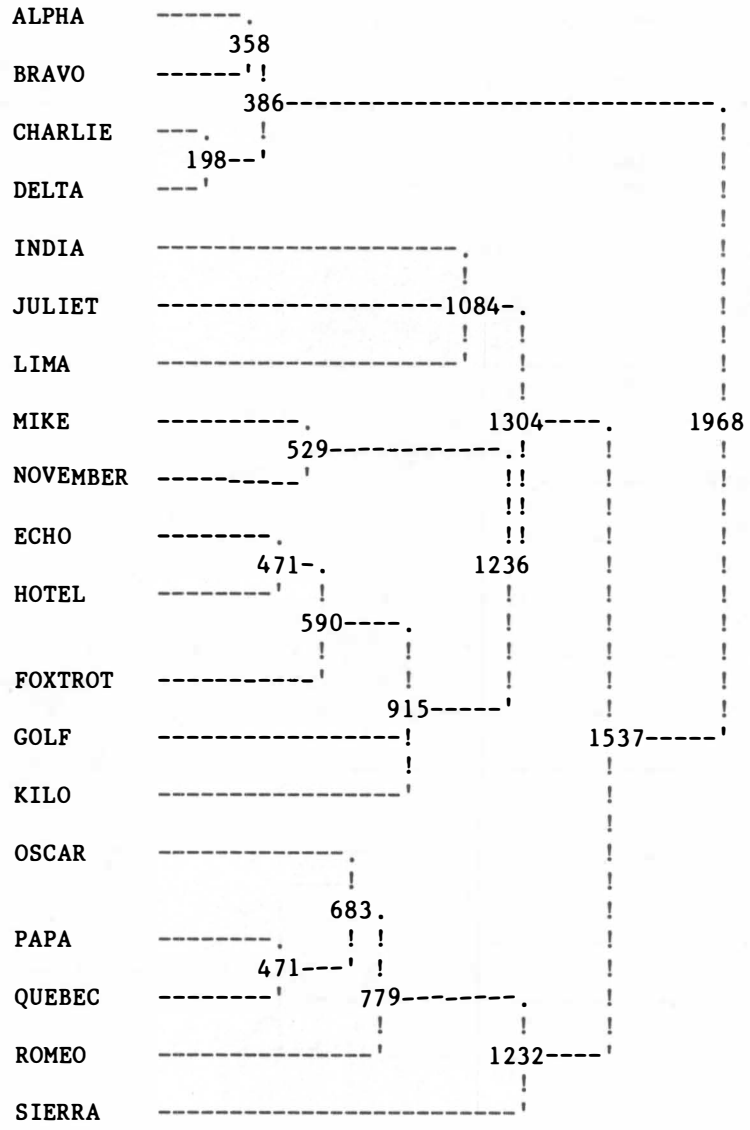
[221]

Second experiment: traditional lexicostatistical techniques, minimum-percentage method, no tolerance



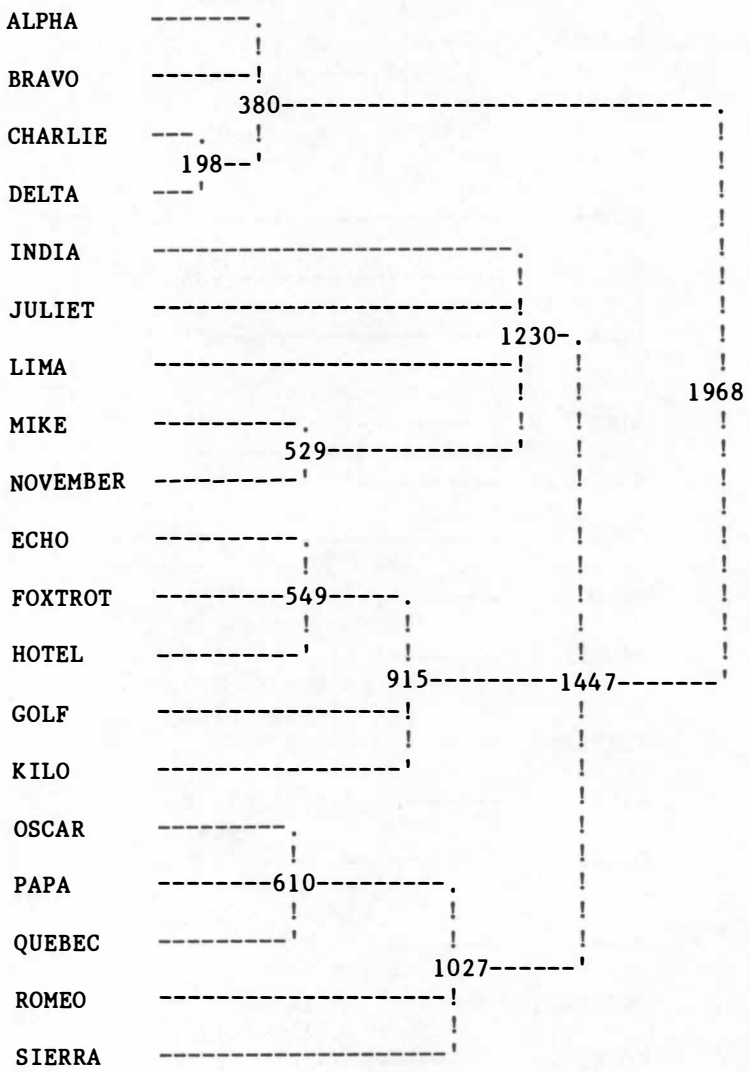
[222]

Second experiment: traditional lexicostatistical techniques, minimum-percentage method, 0.5 confidence level



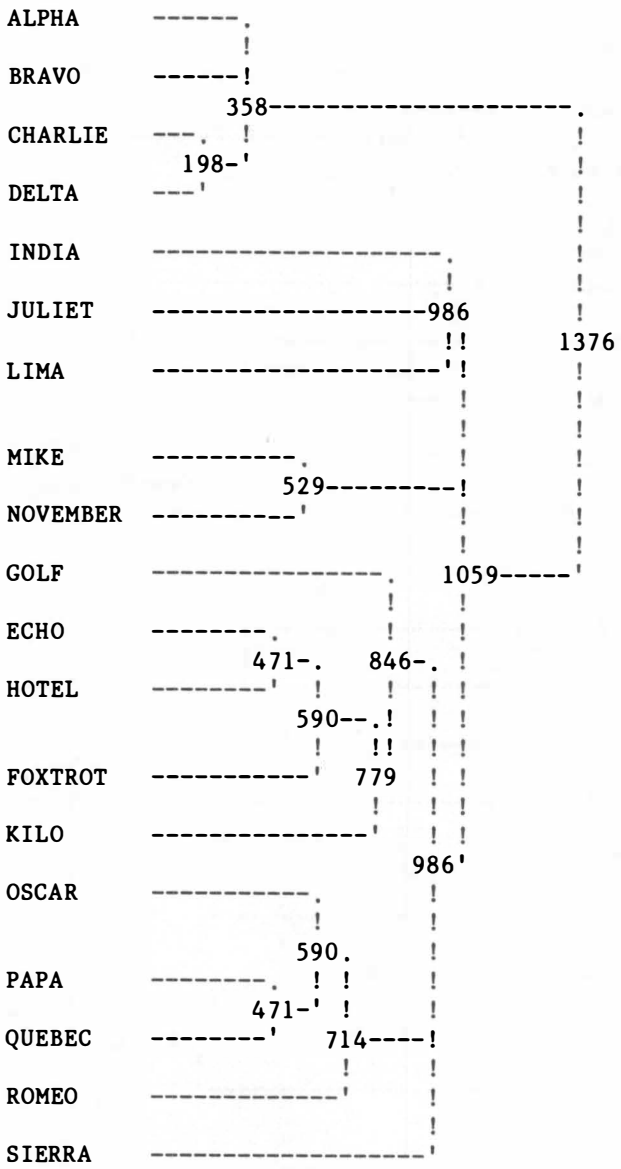
[223]

Second experiment: traditional lexicostatistical techniques, mean-percentage method, no tolerance



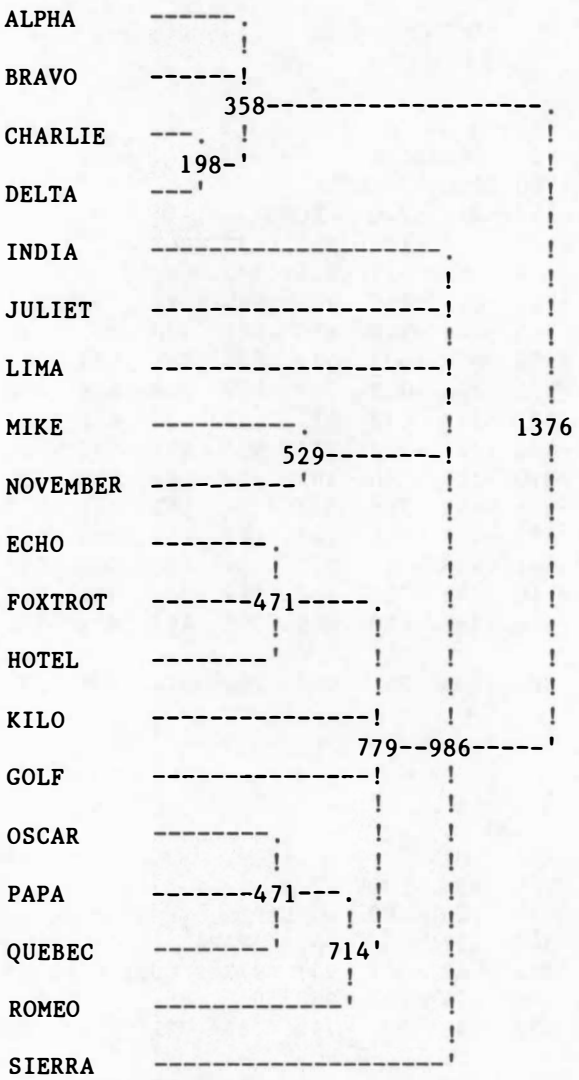
[224]

Second experiment: traditional lexicostatistical techniques, mean-percentage method, 0.5 confidence level



[225]

Second experiment: traditional lexicostatistical techniques, maximum-percentage method, no tolerance



[226]

Second experiment: traditional lexicostatistical techniques, maximum-percentage method, 0.5 confidence level

ALPHA

BRAVO	988	BRAVO									
CHARLIE	983	990	CHARLIE								
DELTA	982	990	1000	DELTA							
ECHO	-241	-257	-218	-240	ECHO						
FOXTROT	-211	-175	-163	-184	953	FOXTROT					
GOLF	-12	6	51	21	923	902	GOLF				
HOTEL	-293	-261	-245	-268	971	982	891	HOTEL			
INDIA	-577	-544	-506	-520	658	701	470	707	INDIA		
JULIET	-711	-666	-628	-639	539	554	446	579	949	JULIET	
KILO	-356	-320	-298	-322	800	849	778	849	860	811	KILO
LIMA	-744	-691	-668	-678	621	697	437	651	947	948	LIMA
MIKE	-472	-408	-376	-393	570	603	491	612	759	732	MIKE
NOVEMBER	-385	-370	-320	-340	493	588	506	543	695	694	NOVEMBER
OSCAR	-639	-686	-693	-706	322	219	162	261	320	321	OSCAR
PAPA	-610	-657	-672	-682	232	162	57	204	221	283	PAPA
QUEBEC	-483	-541	-543	-559	219	89	40	115	127	131	QUEBEC
ROMEO	-713	-716	-738	-751	410	249	250	314	327	376	ROMEO
SIERRA	-648	-648	-682	-680	666	592	493	601	594	580	SIERRA

ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL

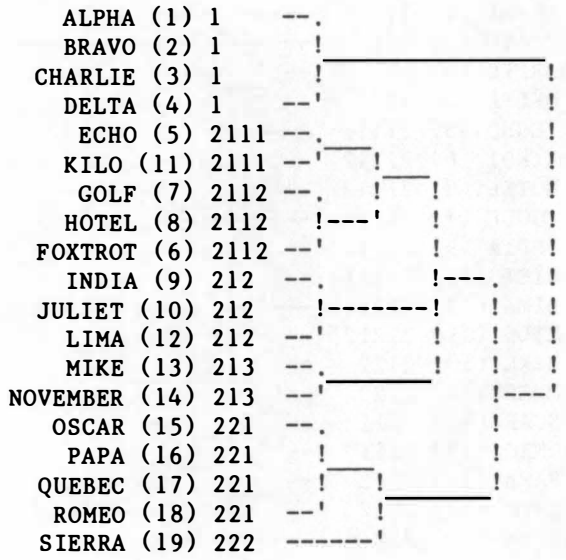
KILO

LIMA	820	LIMA					
MIKE	671	788	MIKE				
NOVEMBER	719	764	964	NOVEMBER			
OSCAR	151	389	126	80	OSCAR		
PAPA	115	300	72	-8	965	PAPA	
QUEBEC	44	234	-42	-107	941	972	QUEBEC
ROMEO	303	475	174	125	963	916	861	ROMEO
SIERRA	542	632	440	379	792	781	721	856	SIERRA

KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

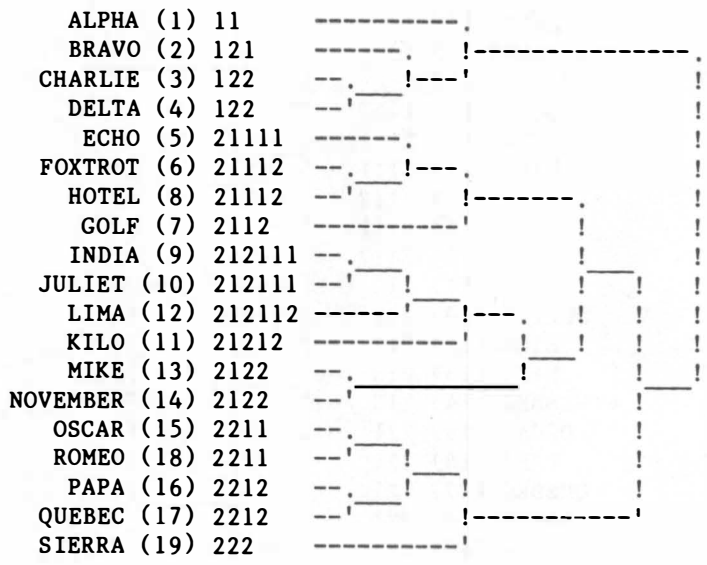
[230]

Second experiment: correlation coefficients



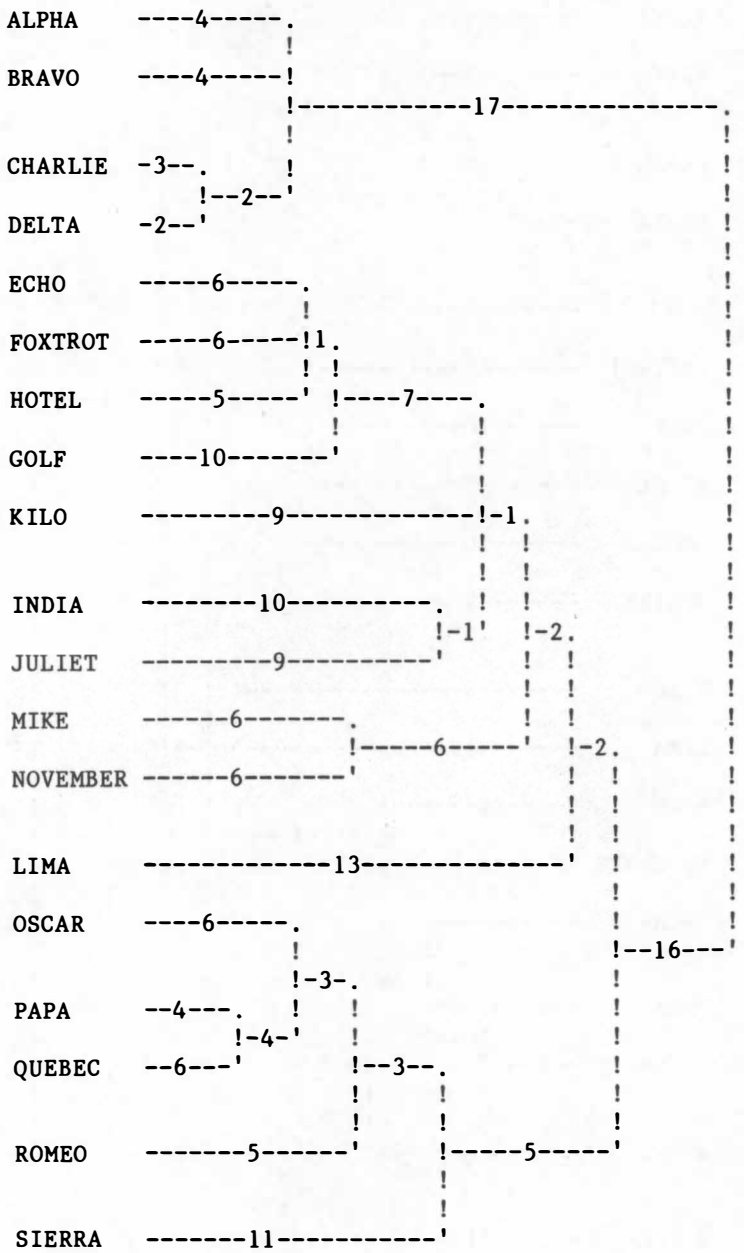
[247]

Second experiment: n-way splitting algorithm, with recomputations



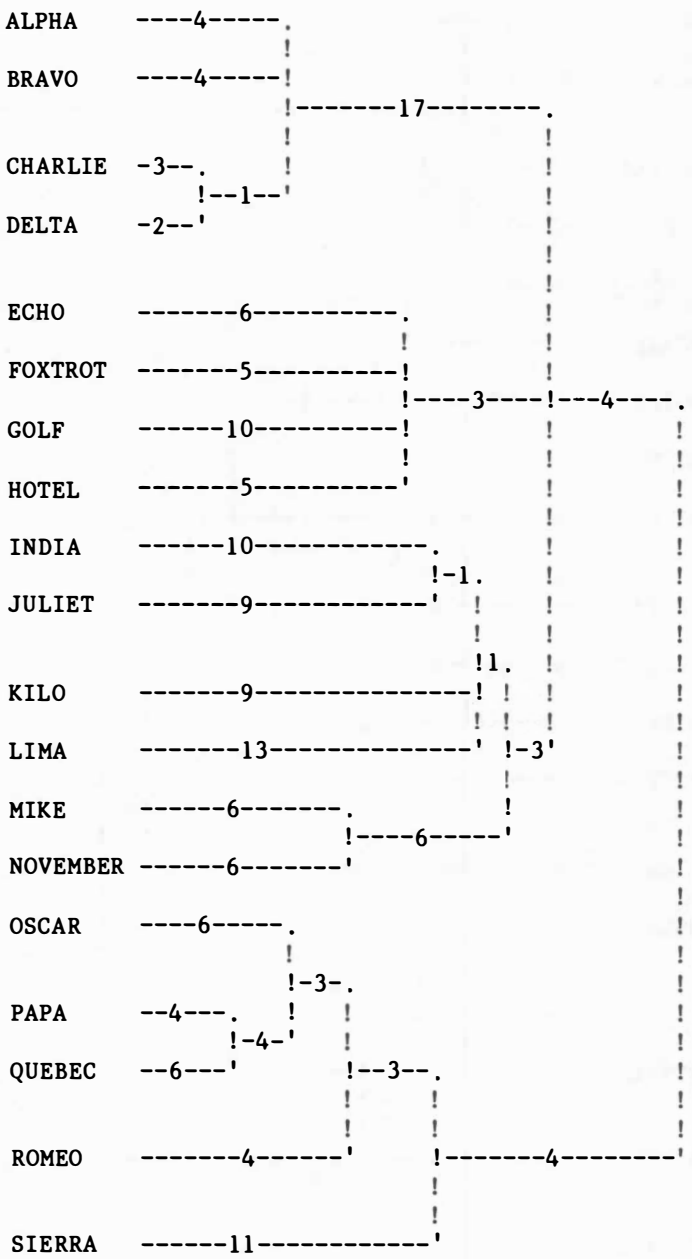
[248]

Second experiment: n-way splitting algorithm, no recomputations



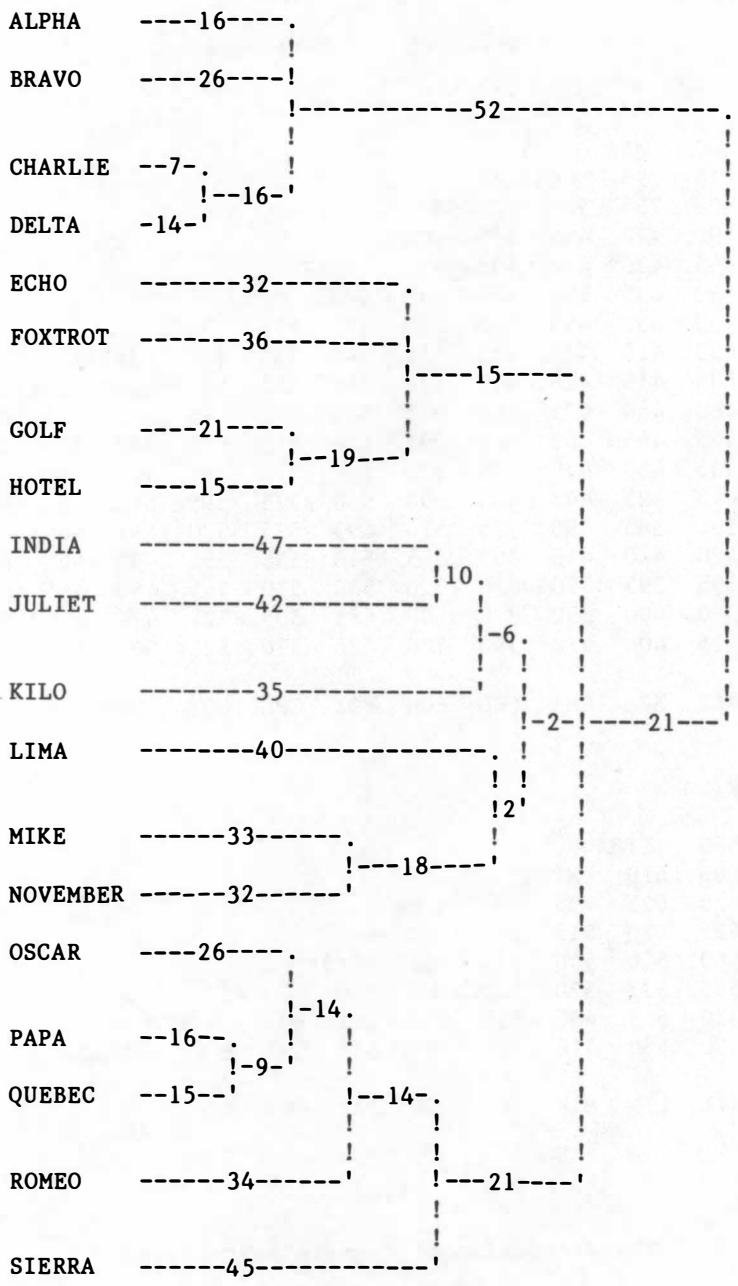
[250]

Second experiment: reduced mutation algorithm



[260]

Second experiment: wild-card algorithm



[300]

Third experiment: 200-item wordlists, 2000-year time depth

ALPHA

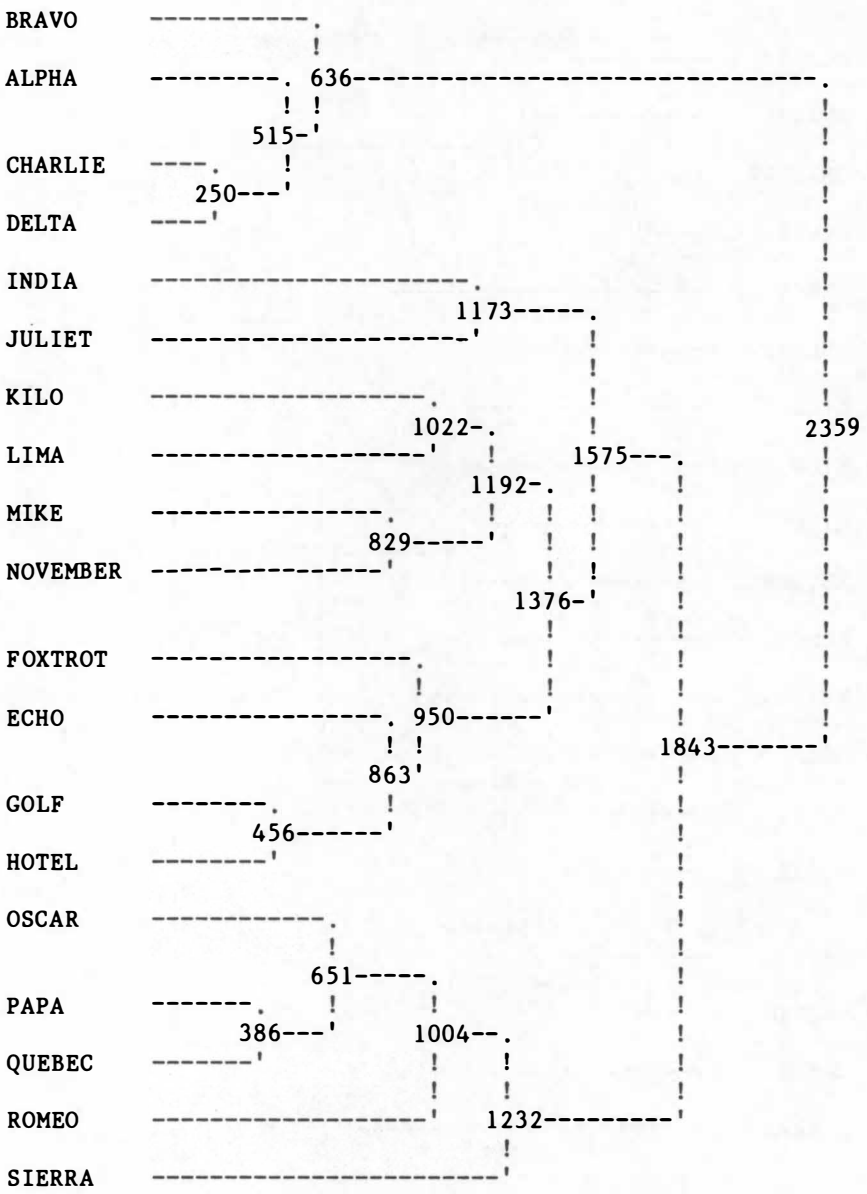
BRAVO	800	BRAVO									
CHARLIE	815	795	CHARLIE								
DELTA	805	765	900	DELTA							
ECHO	490	470	495	475	ECHO						
FOXTROT	445	430	450	435	685	FOXTROT					
GOLF	450	435	455	440	695	670	GOLF				
HOTEL	455	450	465	445	700	670	825	HOTEL			
INDIA	425	410	420	415	555	545	535	550	INDIA		
JULIET	435	410	420	410	570	540	545	550	610	JULIET	
KILO	480	450	475	465	615	585	610	615	600	615	KILO
LIMA	465	465	465	460	595	605	610	615	590	570	LIMA
MIKE	455	450	470	445	580	560	570	585	515	530	MIKE
NOVEMBER	445	425	445	430	590	560	575	595	545	565	NOVEMBER
OSCAR	390	385	390	375	510	495	535	530	465	465	OSCAR
PAPA	420	410	415	395	535	540	535	555	490	485	PAPA
QUEBEC	395	395	390	370	520	530	530	545	480	465	QUEBEC
ROMEO	390	400	390	370	500	485	505	525	470	460	ROMEO
SIERRA	415	400	410	390	520	525	520	525	500	495	SIERRA
	ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL	

KILO

LIMA	650	LIMA						
MIKE	605	610	MIKE					
NOVEMBER	625	625	705	NOVEMBER				
OSCAR	525	520	515	520	OSCAR			
PAPA	560	540	540	535	765	PAPA		
QUEBEC	555	535	520	520	760	850	QUEBEC	
ROMEO	530	555	485	510	655	695	695	ROMEO
SIERRA	550	550	510	525	595	625	595	600
	KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

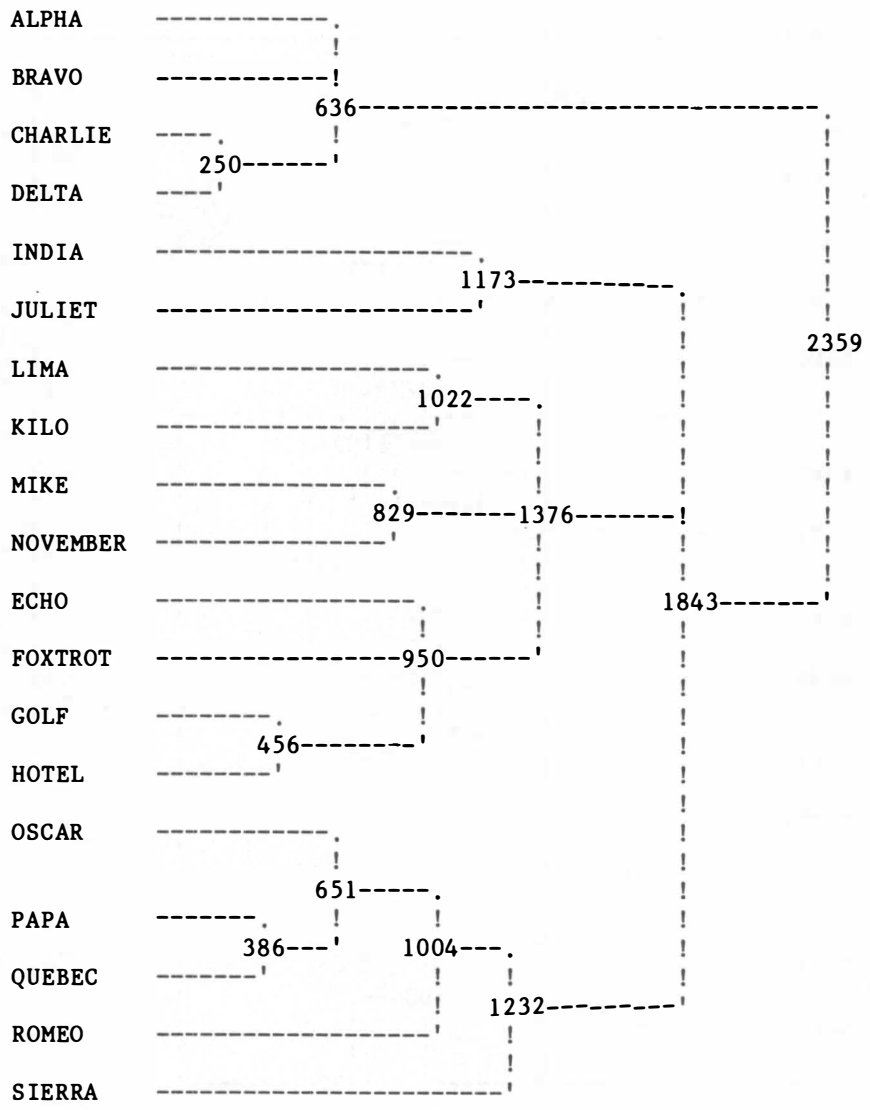
[310]

Third experiment: cognate percentages



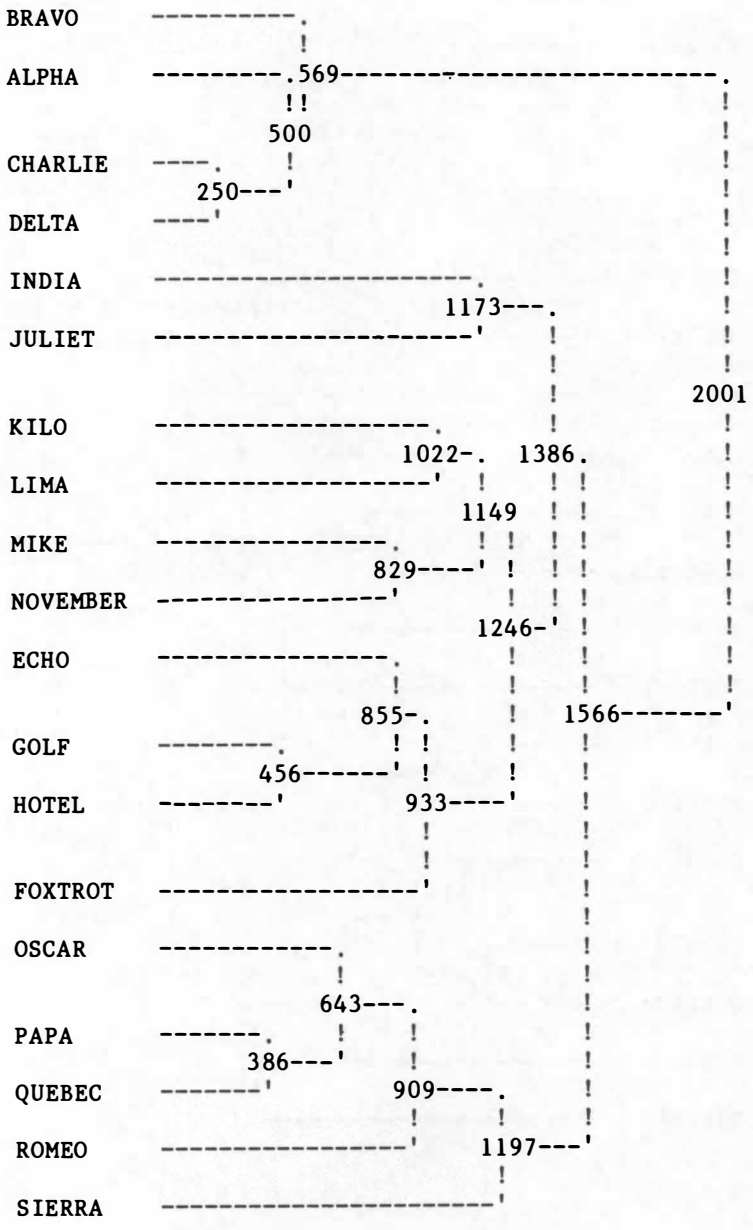
[321]

Third experiment: traditional lexicostatistical techniques, minimum-percentage method, no tolerance



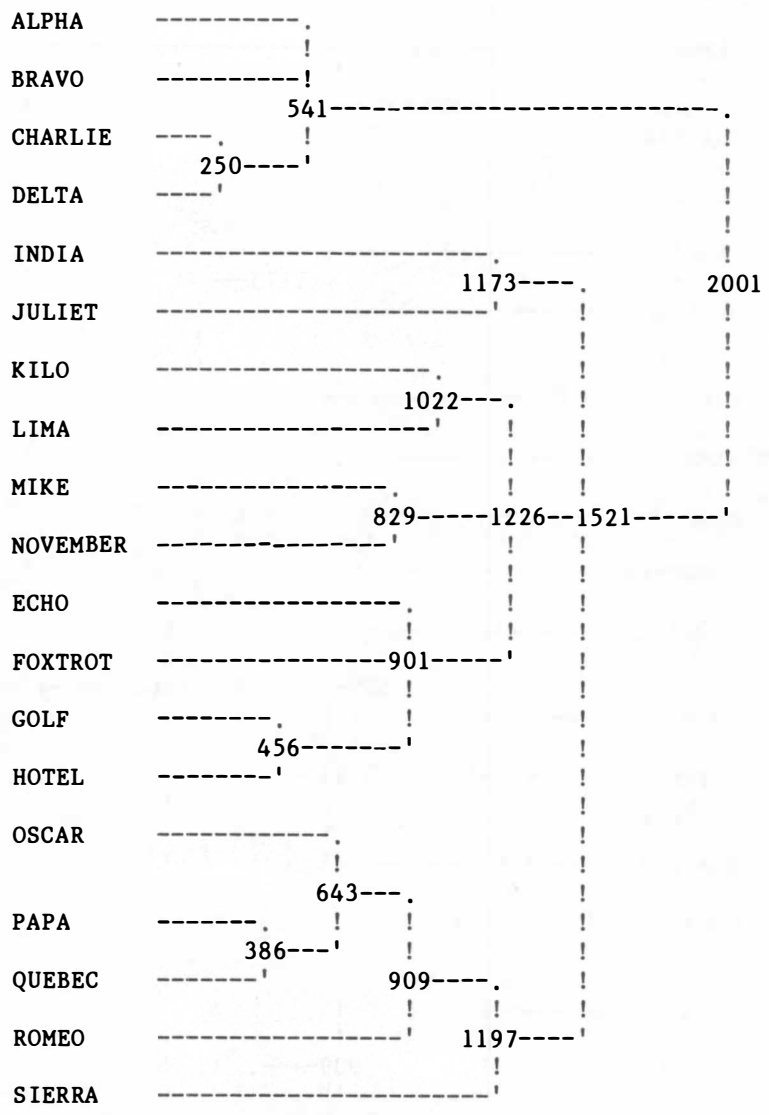
[322]

Third experiment: traditional lexicostatistical techniques, minimum-percentage method, 0.5 confidence level



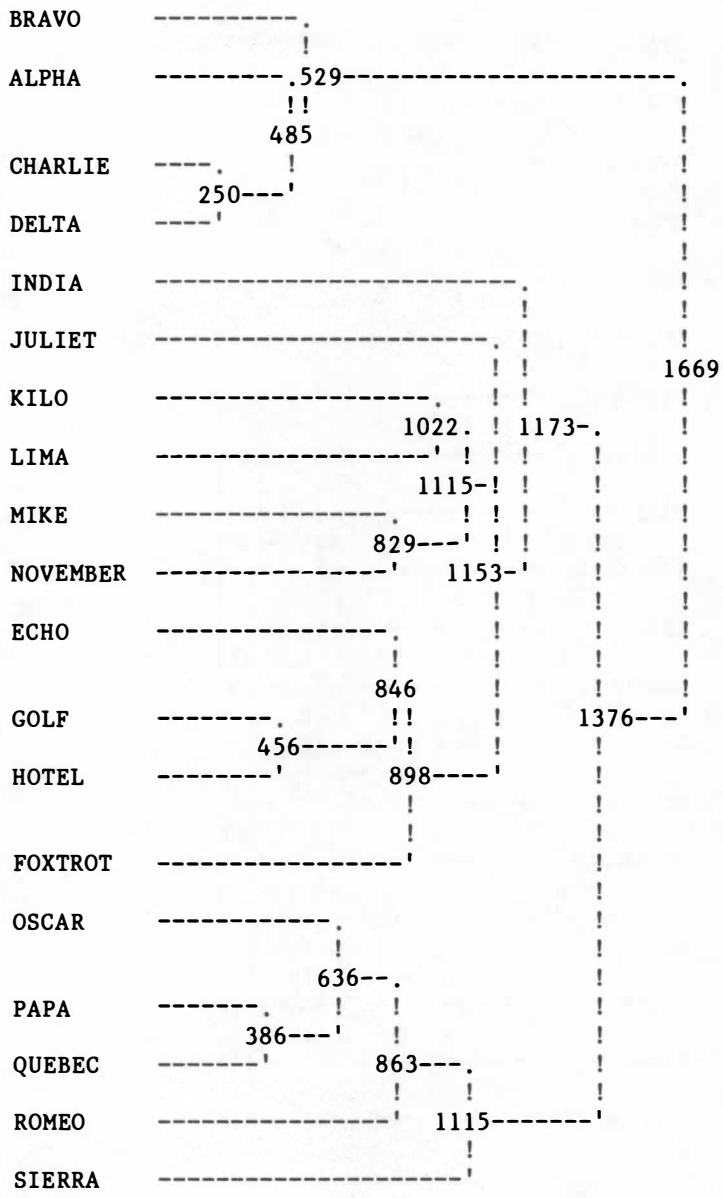
[323]

Third experiment: traditional lexicostatistical techniques, mean-percentage method, no tolerance



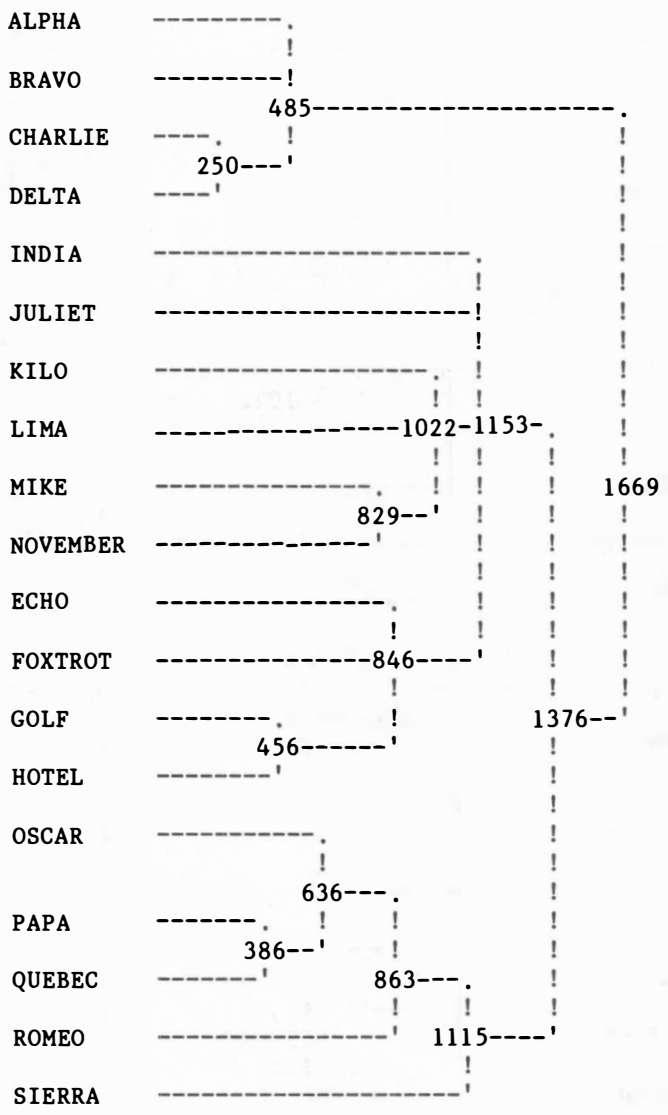
[324]

Third experiment: traditional lexicostatistical techniques, mean-percentage method, 0.5 confidence level



[325]

Third experiment: traditional lexicostatistical techniques, maximum-percentage method, no tolerance



[326]

Third experiment: traditional lexicostatistical techniques, maximum-percentage method, 0.5 confidence level

ALPHA

BRAVO	997	BRAVO									
CHARLIE	991	986	CHARLIE								
DELTA	988	991	999	DELTA							
ECHO	-434	-422	-430	-389	ECHO						
FOXTROT	-538	-535	-534	-509	967	FOXTROT					
GOLF	-473	-465	-464	-446	945	939	GOLF				
HOTEL	-493	-504	-494	-466	939	945	995	HOTEL			
INDIA	-612	-625	-615	-590	741	779	706	676	INDIA		
JULIET	-583	-581	-570	-536	735	794	692	695	985	JULIET	
KILO	-714	-699	-703	-676	762	856	738	745	953	938	KILO
LIMA	-695	-716	-689	-676	814	827	759	769	894	926	LIMA
MIKE	-512	-525	-532	-481	688	727	664	654	781	786	MIKE
NOVEMBER	-629	-617	-617	-594	681	760	678	655	810	804	NOVEMBER
OSCAR	-699	-671	-695	-691	107	292	206	248	214	150	OSCAR
PAPA	-696	-661	-687	-677	38	182	194	173	142	86	PAPA
QUEBEC	-681	-660	-674	-665	92	219	217	212	181	153	QUEBEC
ROMEO	-730	-716	-731	-718	124	320	264	243	264	212	ROMEO
SIERRA	-882	-855	-877	-861	298	442	407	417	453	405	SIERRA

ALP BRA CHA DEL ECH FOX GOL HOT IND JUL

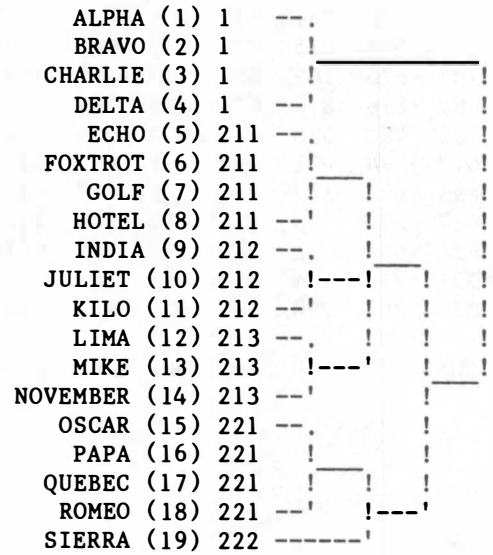
KILO

LIMA	959	LIMA					
MIKE	847	854	MIKE				
NOVEMBER	894	902	972	NOVEMBER			
OSCAR	325	318	263	297	OSCAR		
PAPA	220	252	180	245	992	PAPA	
QUEBEC	246	283	240	287	991	997	QUEBEC
ROMEO	351	294	340	332	982	968	969	ROMEO
SIERRA	528	534	471	512	919	883	923	949	SIERRA

KIL LIM MIK NOV OSC PAP QUE ROM

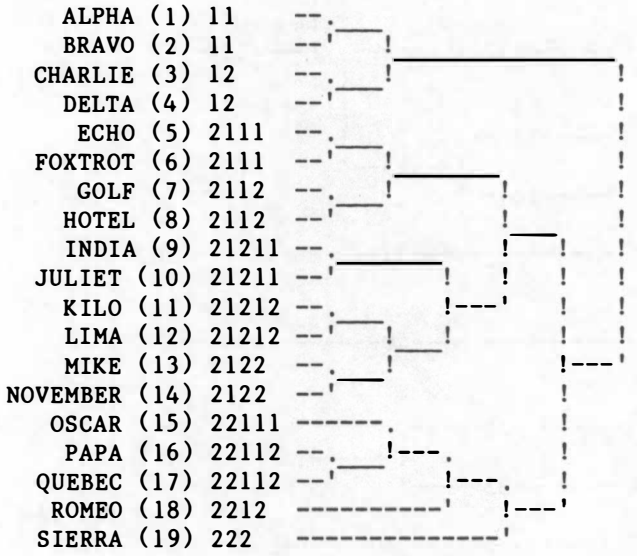
[330]

Third experiment: correlation coefficients



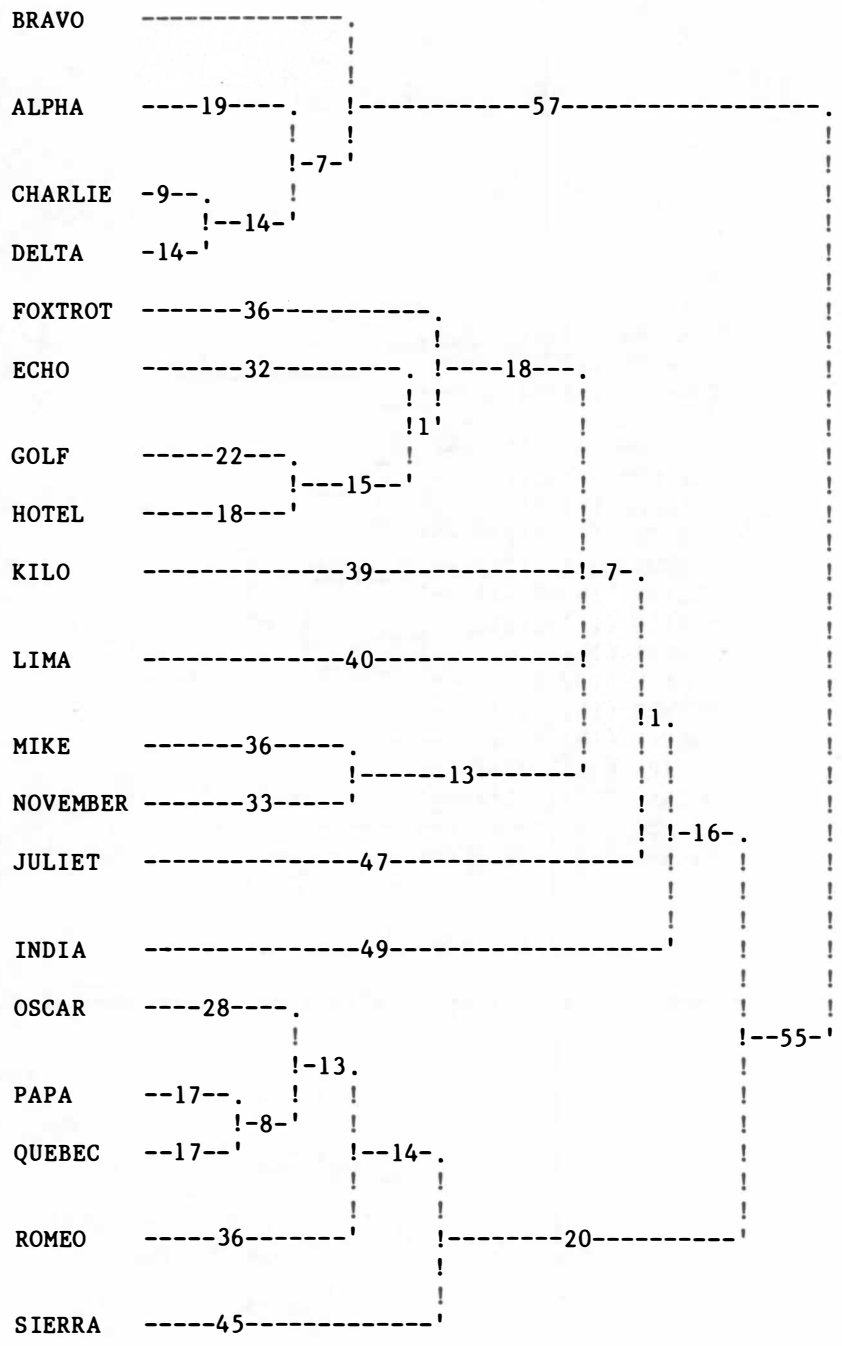
[347]

Third experiment: n-way splitting algorithm, with recomputations



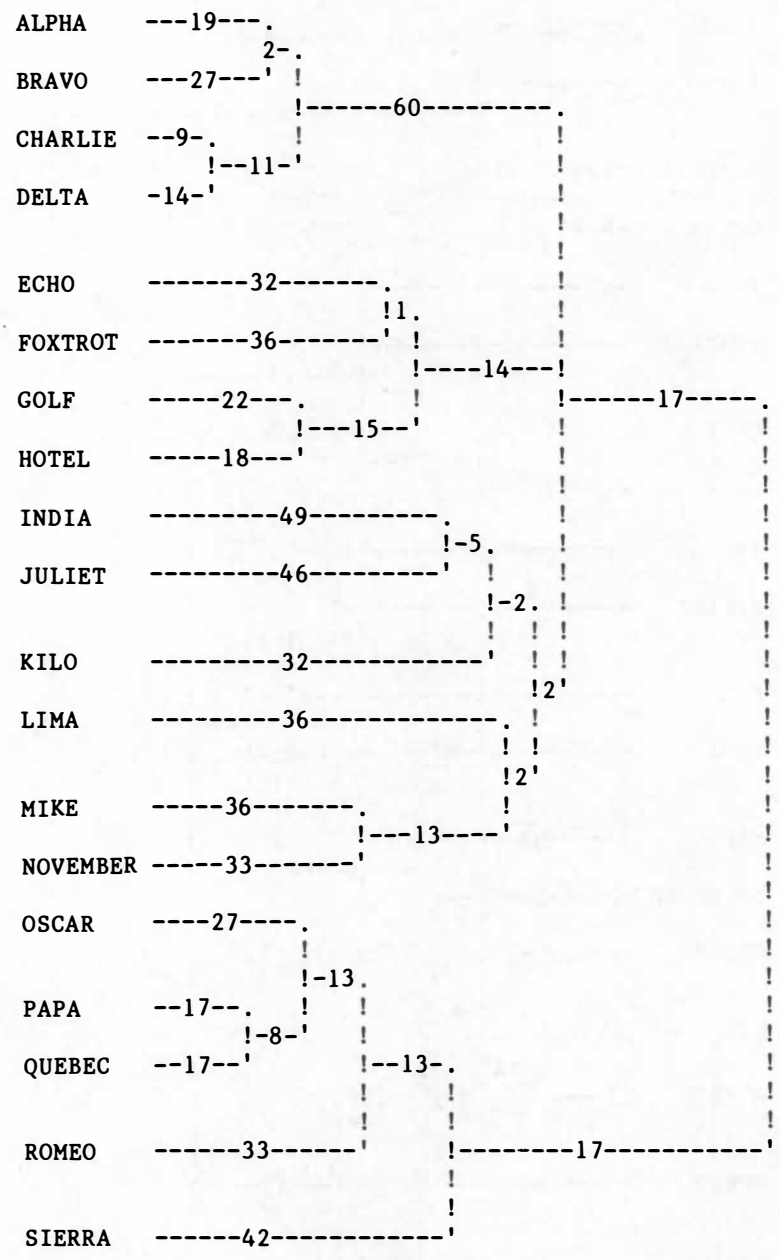
[348]

Third experiment: n-way splitting algorithm, no recomputations



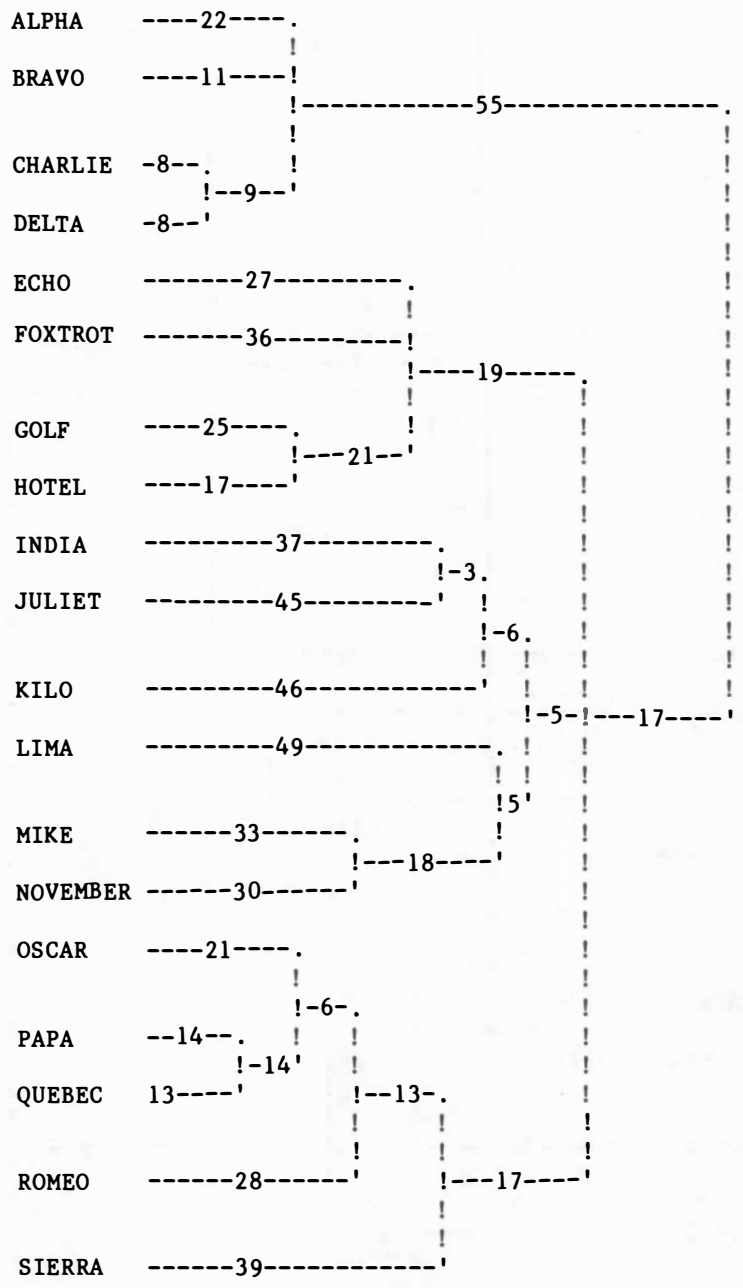
[350]

Third experiment: reduced mutation algorithm



[360]

Third experiment: wild-card algorithm



[400]

Fourth experiment: 200-items wordlists, 2000-year time depth

ALPHA

BRAVO	845									
CHARLIE	810	875								
DELTA	810	870	920							
ECHO	440	465	450	460						
FOXTROT	410	425	410	415	715					
GOLF	400	410	405	415	655	650				
HOTEL	430	435	435	435	720	670	800			
INDIA	415	440	450	435	595	565	560	595		
JULIET	430	455	450	440	565	535	525	540	635	
KILO	450	475	460	450	555	535	495	525	625	600
LIMA	400	425	435	425	575	550	530	560	615	550
MIKE	405	425	430	420	560	510	510	535	570	540
NOVEMBER	415	445	455	430	580	525	530	575	600	570
OSCAR	415	420	405	410	570	540	525	530	570	540
PAPA	395	395	380	385	535	520	515	515	540	510
QUEBEC	410	420	395	395	525	520	505	505	550	540
ROMEO	440	440	425	430	550	545	530	535	550	555
SIERRA	445	465	455	445	570	515	500	545	565	515

ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL

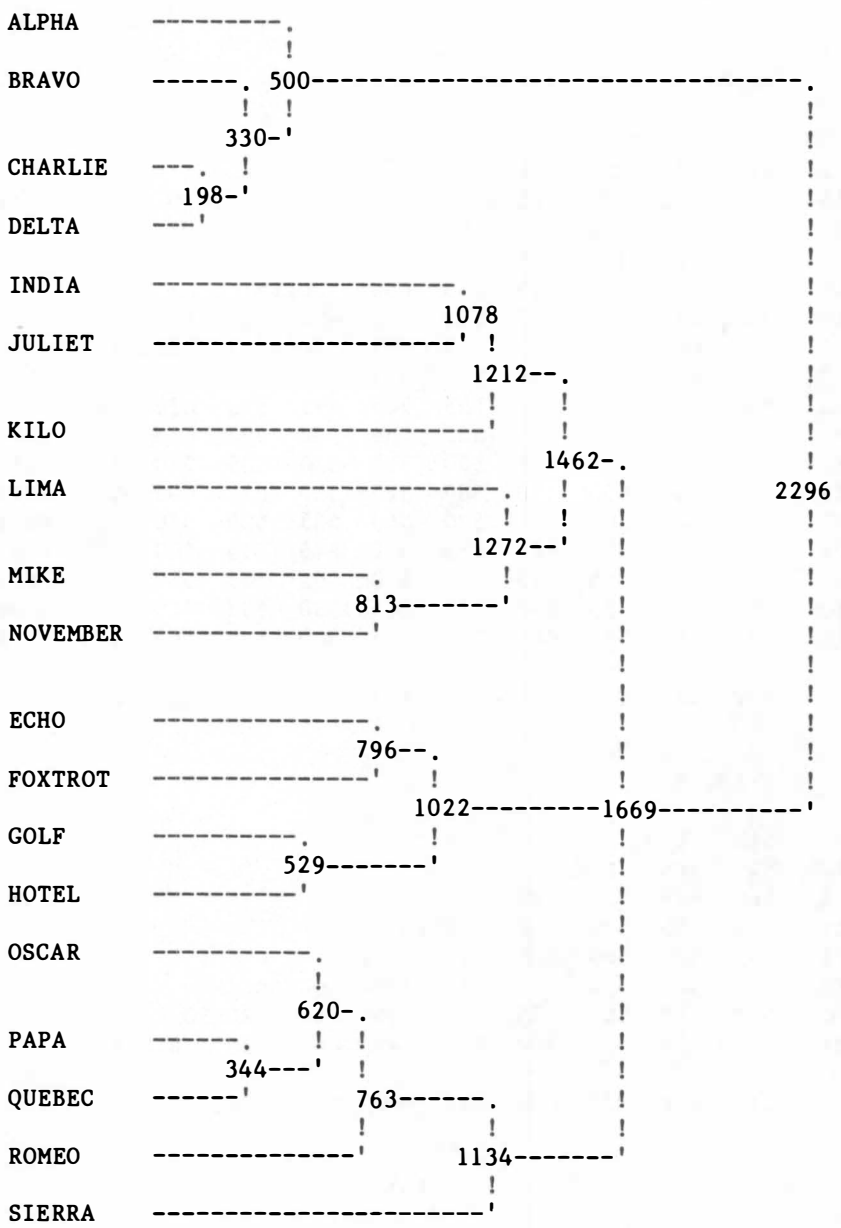
KILO

LIMA	545						
MIKE	540	585					
NOVEMBER	560	595	710				
OSCAR	535	535	530	550			
PAPA	535	505	500	510	770		
QUEBEC	540	505	510	525	780	865	
ROMEO	550	510	500	535	745	725	740
SIERRA	540	530	515	550	660	625	645	620

KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

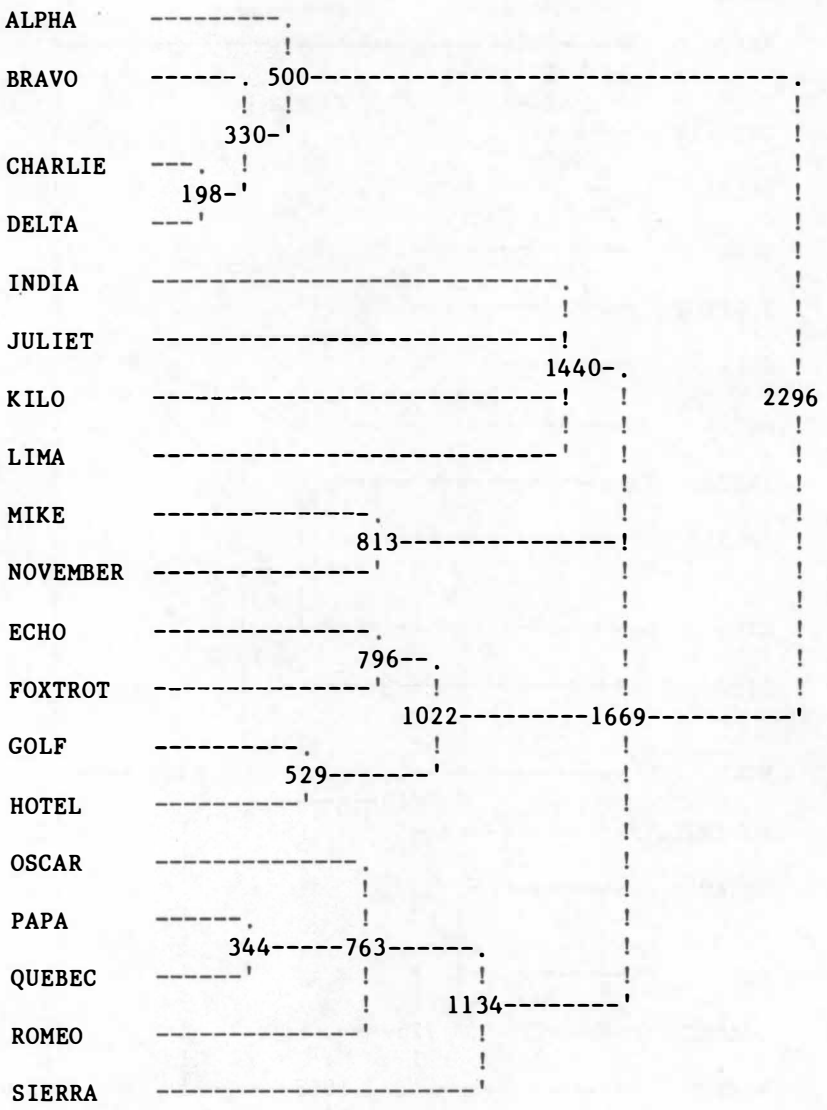
[410]

Fourth experiment: cognate percentages



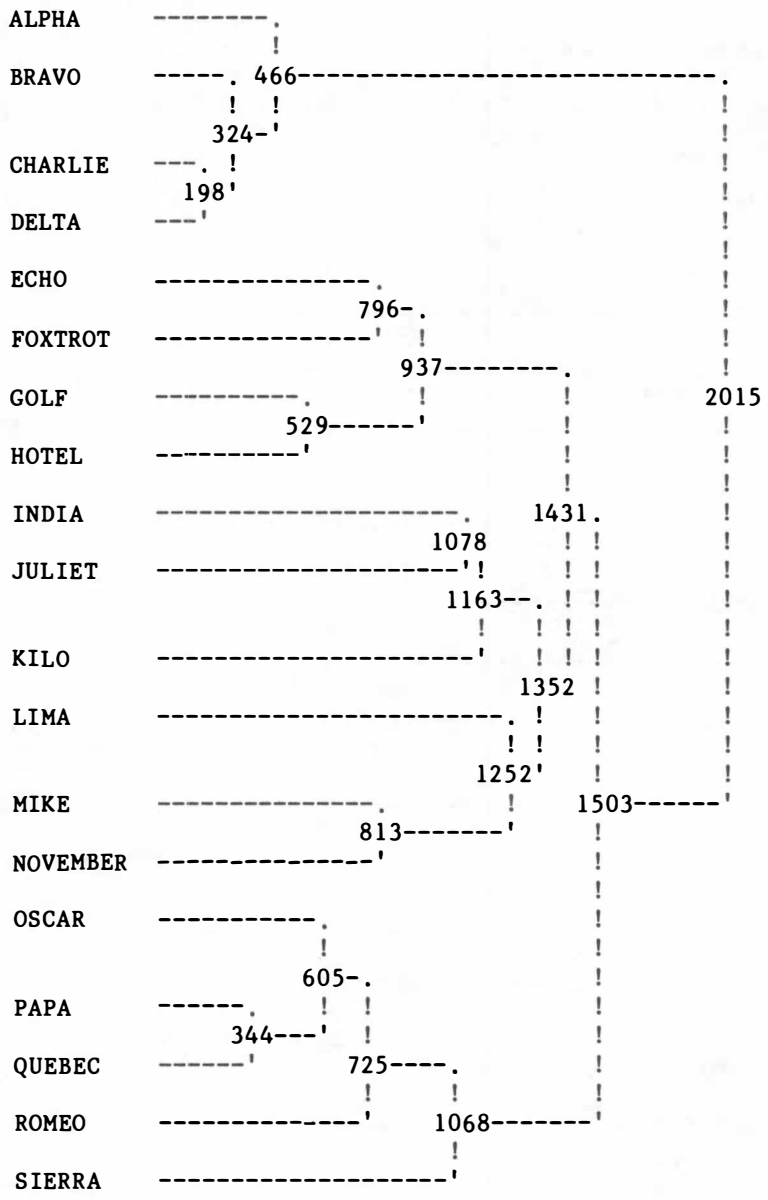
[421]

Fourth experiment: traditional lexicostatistical techniques, minimum-percentage method, no tolerance



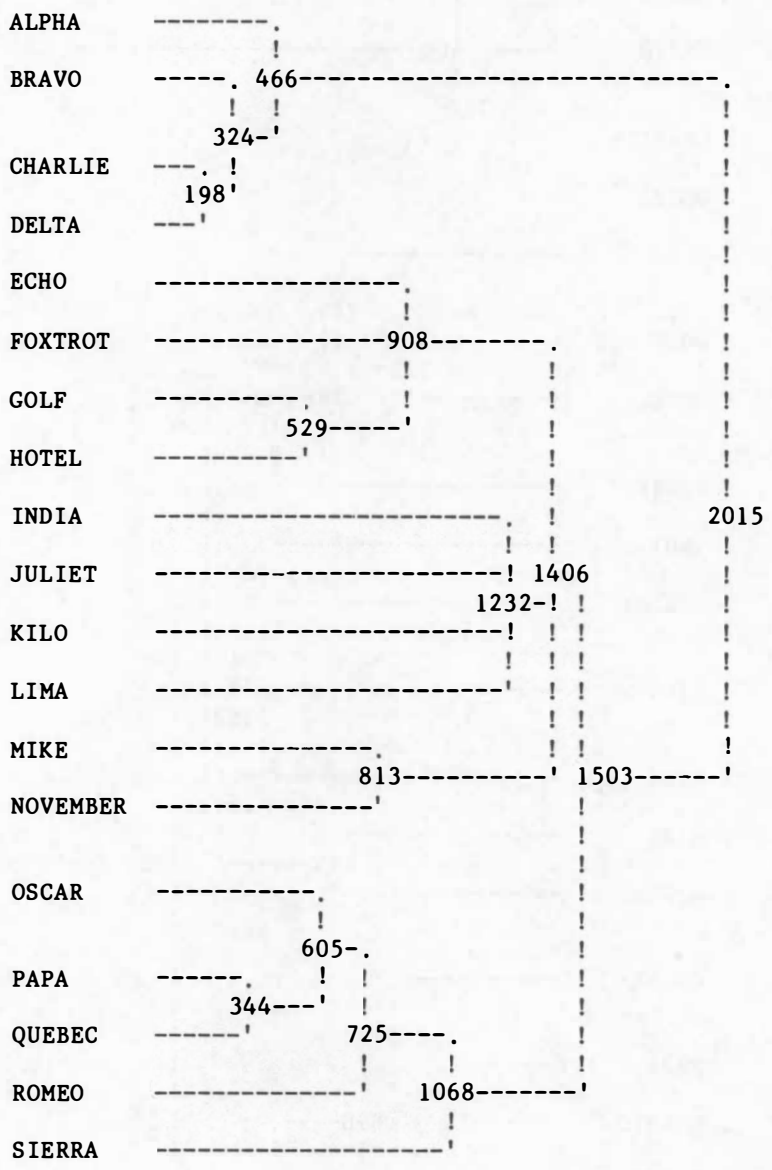
[422]

Fourth experiment: traditional lexicostatistical techniques, minimum-percentage method, 0.5 confidence level



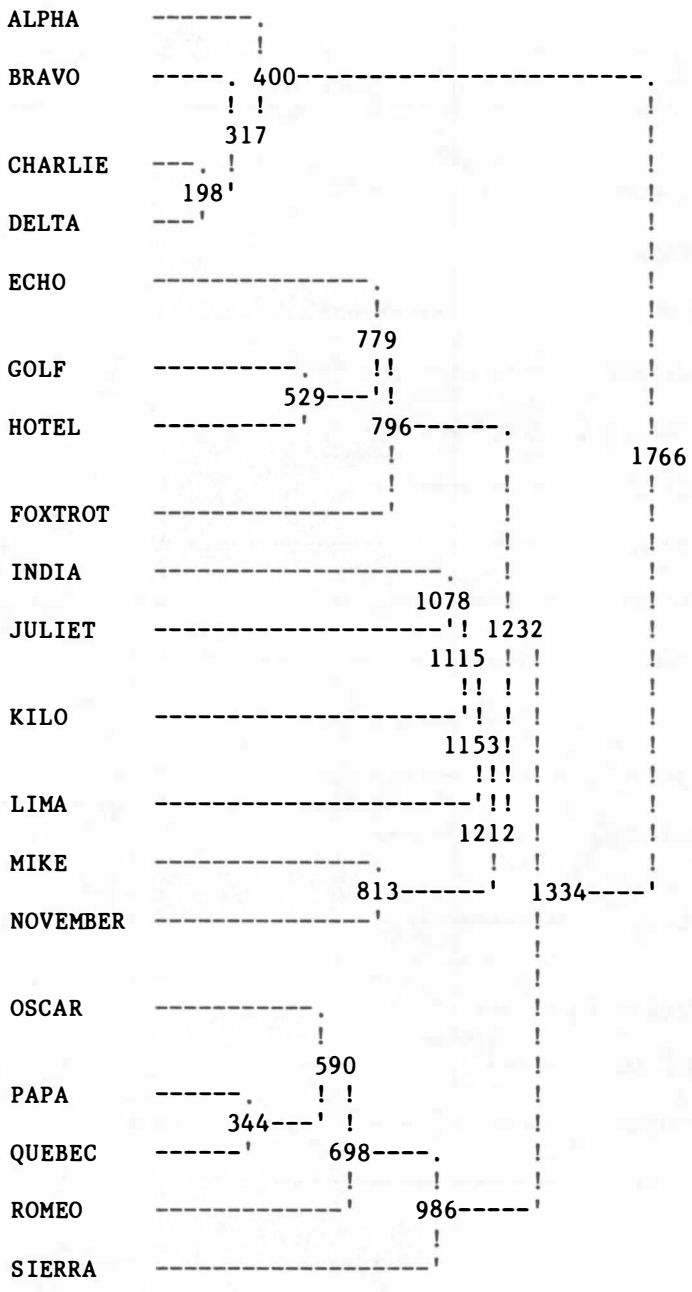
[423]

Fourth experiment: traditional lexicostatistical techniques, mean-percentage method, no tolerance



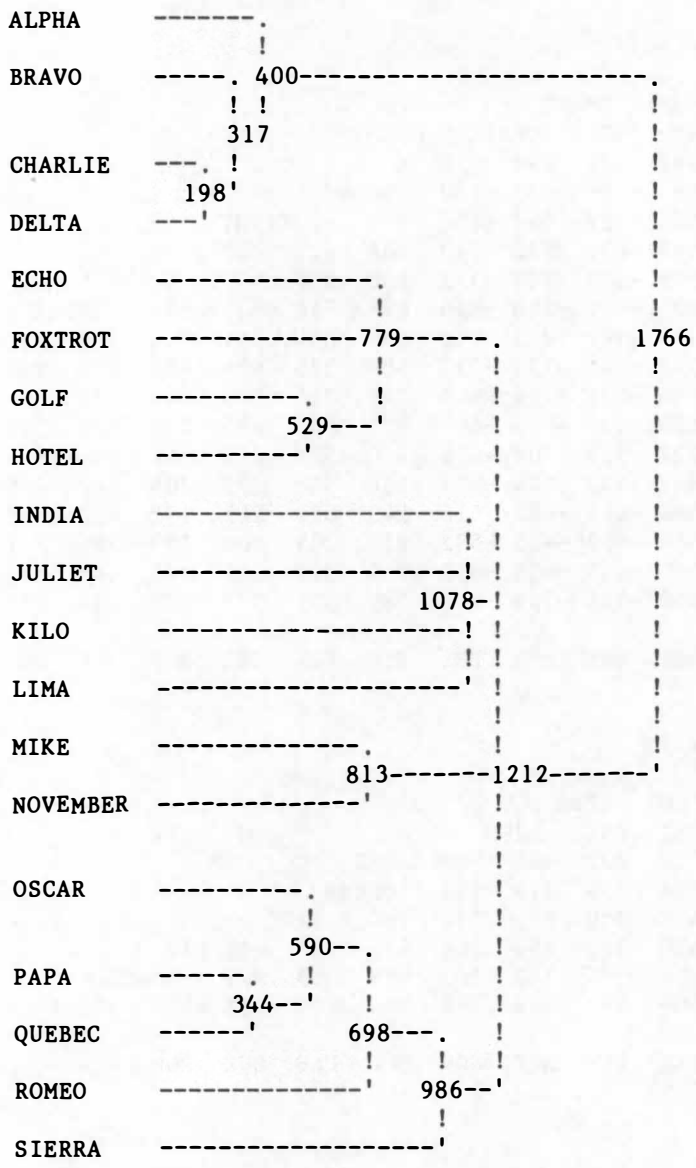
[424]

Fourth experiment: traditional lexicostatistical techniques, mean-percentage method, 0.5 confidence level



[425]

Fourth experiment: traditional lexicostatistical techniques, maximum-percentage method, no tolerance



[426]

Fourth experiment: traditional lexicostatistical techniques, maximum-percentage method, 0.5 confidence level

ALPHA

BRAVO	998	BRAVO									
CHARLIE	989	992	CHARLIE								
DELTA	992	994	998	DELTA							
ECHO	-682	-706	-675	-680	ECHO						
FOXTROT	-684	-699	-681	-668	973	FOXTROT					
GOLF	-592	-607	-592	-590	948	922	GOLF				
HOTEL	-599	-597	-588	-571	900	926	977	HOTEL			
INDIA	-875	-861	-848	-850	688	716	631	570	INDIA		
JULIET	-777	-781	-757	-761	590	654	551	534	959	JULIET	
KILO	-743	-760	-721	-727	489	528	496	407	919	964	KILO
LIMA	-836	-821	-796	-803	729	716	670	640	931	907	LIMA
MIKE	-689	-677	-647	-665	525	550	483	482	812	781	MIKE
NOVEMBER	-727	-729	-709	-696	540	578	515	441	801	755	NOVEMBER
OSCAR	-642	-671	-684	-677	238	348	273	204	442	466	OSCAR
PAPA	-580	-606	-625	-616	243	327	241	172	430	474	PAPA
QUEBEC	-562	-602	-605	-592	215	269	203	133	388	371	QUEBEC
ROMEO	-617	-643	-658	-650	259	322	256	175	448	407	ROMEO
SIERRA	-660	-706	-712	-690	289	451	379	209	524	598	SIERRA

ALP BRA CHA DEL ECH FOX GOL HOT IND JUL

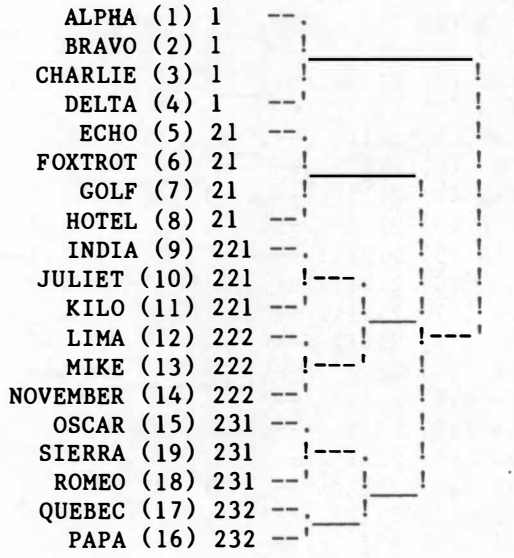
KILO

LIMA	860	LIMA						
MIKE	731	881	MIKE					
NOVEMBER	728	902	987	NOVEMBER				
OSCAR	536	376	319	319	OSCAR			
PAPA	458	349	291	314	980	PAPA		
QUEBEC	450	326	256	264	976	994	QUEBEC	
ROMEO	466	366	312	266	989	983	977	ROMEO
SIERRA	586	487	448	393	964	944	918	954	SIERRA

KIL LIM MIK NOV OSC PAP QUE ROM

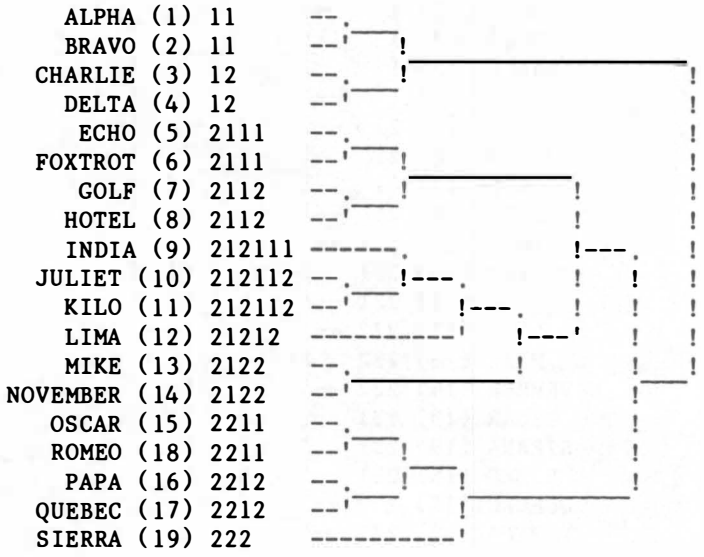
[430]

Fourth experiment: correlation coefficients



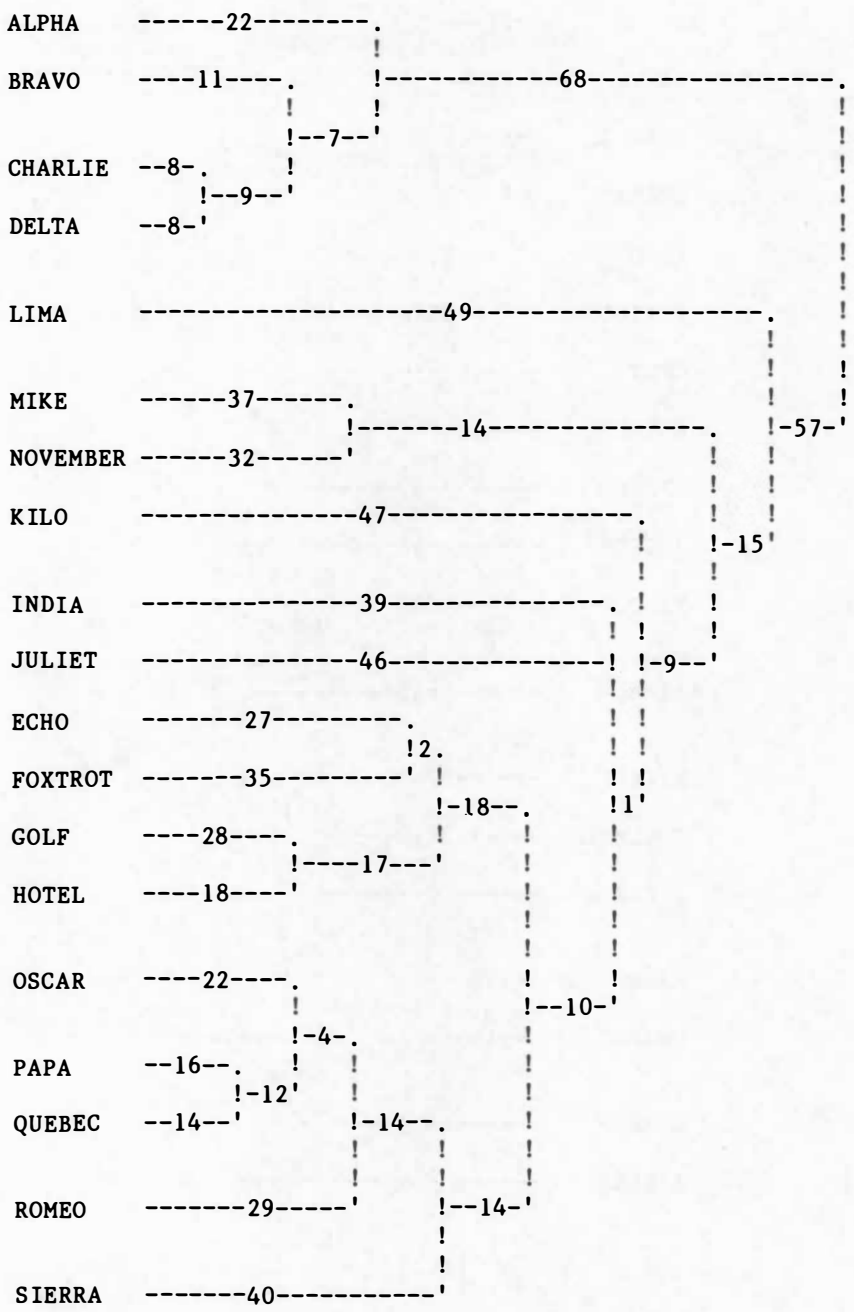
[447]

Fourth experiment: n-way splitting algorithm, with recomputations



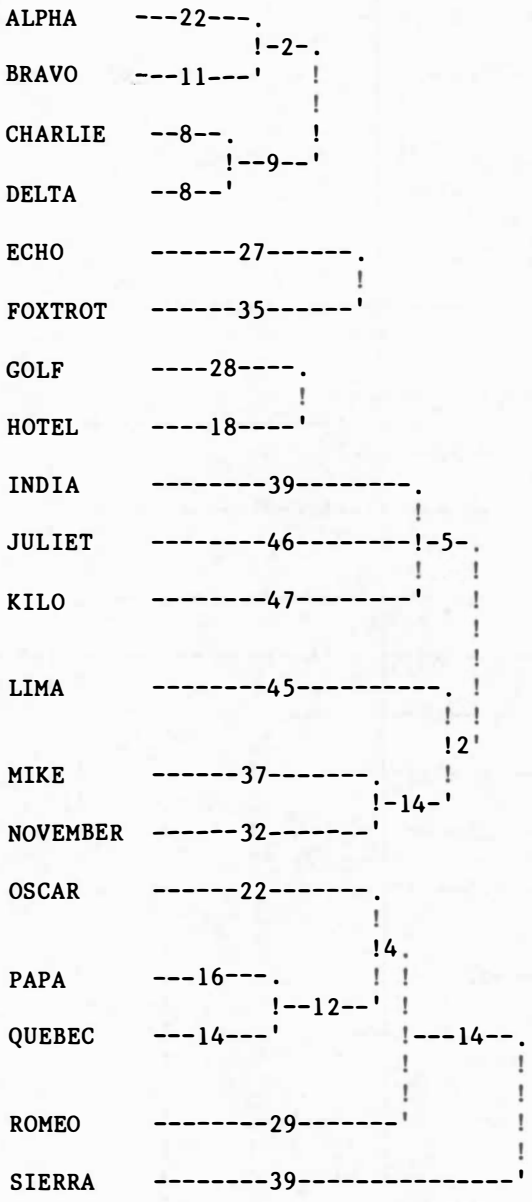
[448]

Fourth experiment: n-way splitting algorithm, no recomputations



[450]

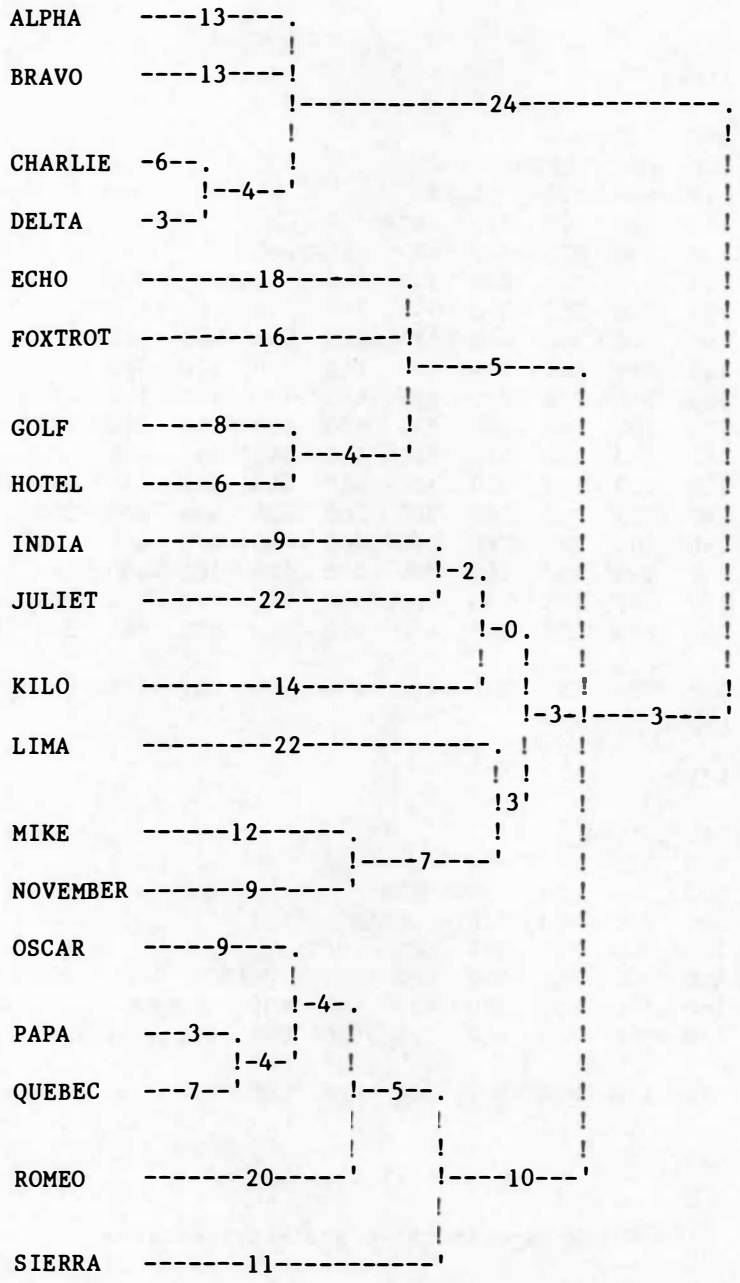
Fourth experiment: reduced mutation algorithm



[460]

Fourth experiment: wild-card algorithm

The reconstitution is incomplete: the algorithm was stopped by the presence of chance cognate-like forms.



[500]

Fifth experiment: 50-item wordlists, 4000-year time depth

ALPHA

BRAVO	600									
CHARLIE	620	600								
DELTA	660	660	820							
ECHO	160	180	200	220						
FOXTROT	260	300	320	340	420					
GOLF	260	240	240	260	500	540				
HOTEL	280	260	280	300	540	520	740			
INDIA	300	320	360	340	420	420	520	500		
JULIET	180	200	200	180	280	260	320	340	480	
KILO	240	240	280	280	400	400	480	480	540	420
LIMA	220	200	240	240	240	280	300	340	360	220
MIKE	260	280	300	300	240	360	360	360	460	340
NOVEMBER	260	320	340	320	280	440	400	420	540	360
OSCAR	180	220	200	180	300	260	420	440	400	300
PAPA	260	280	280	280	360	360	440	520	460	320
QUEBEC	220	240	260	260	360	340	420	500	400	280
ROMEO	160	180	160	180	220	300	340	320	300	200
SIERRA	260	260	260	260	440	420	460	500	440	340

ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL

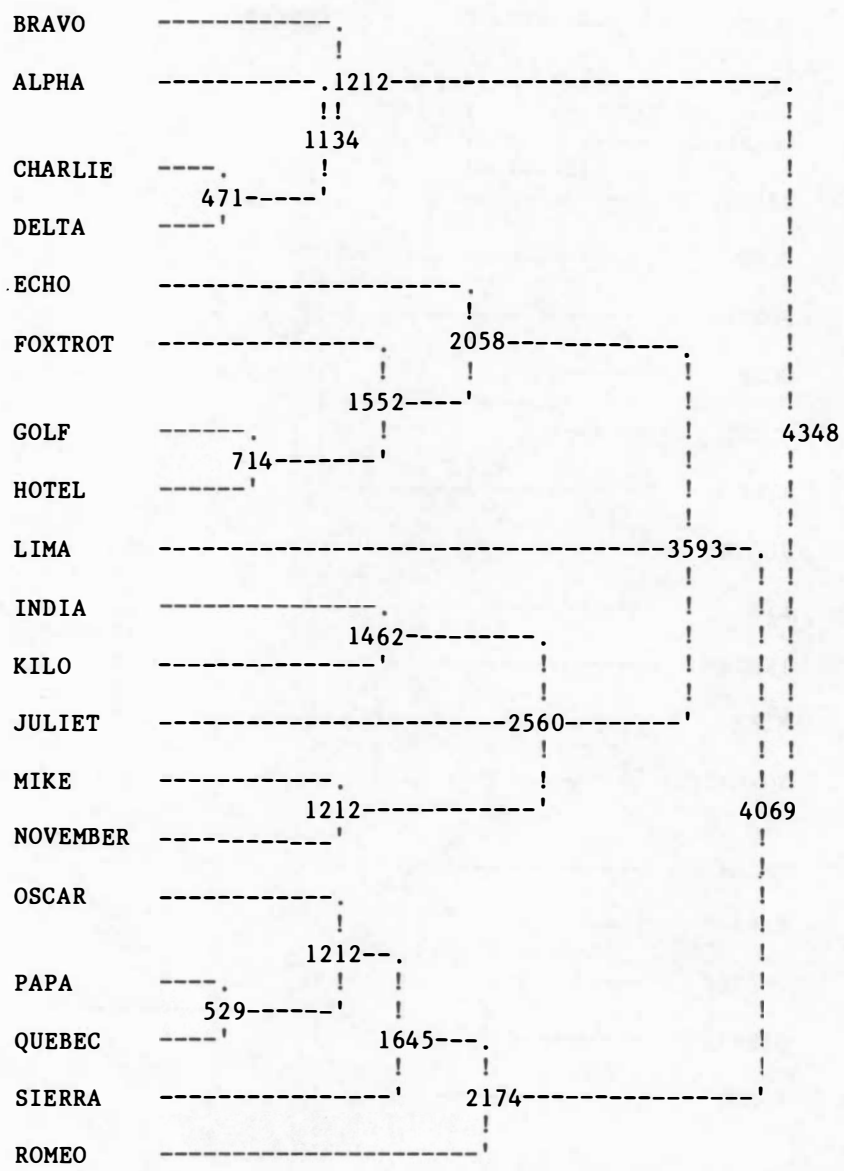
KILO

LIMA	360						
MIKE	420	420					
NOVEMBER	500	380	600				
OSCAR	340	240	300	320			
PAPA	400	340	320	380	680		
QUEBEC	400	360	340	380	600	800	
ROMEO	280	180	260	280	420	460	400
SIERRA	460	300	300	400	500	620	540

KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

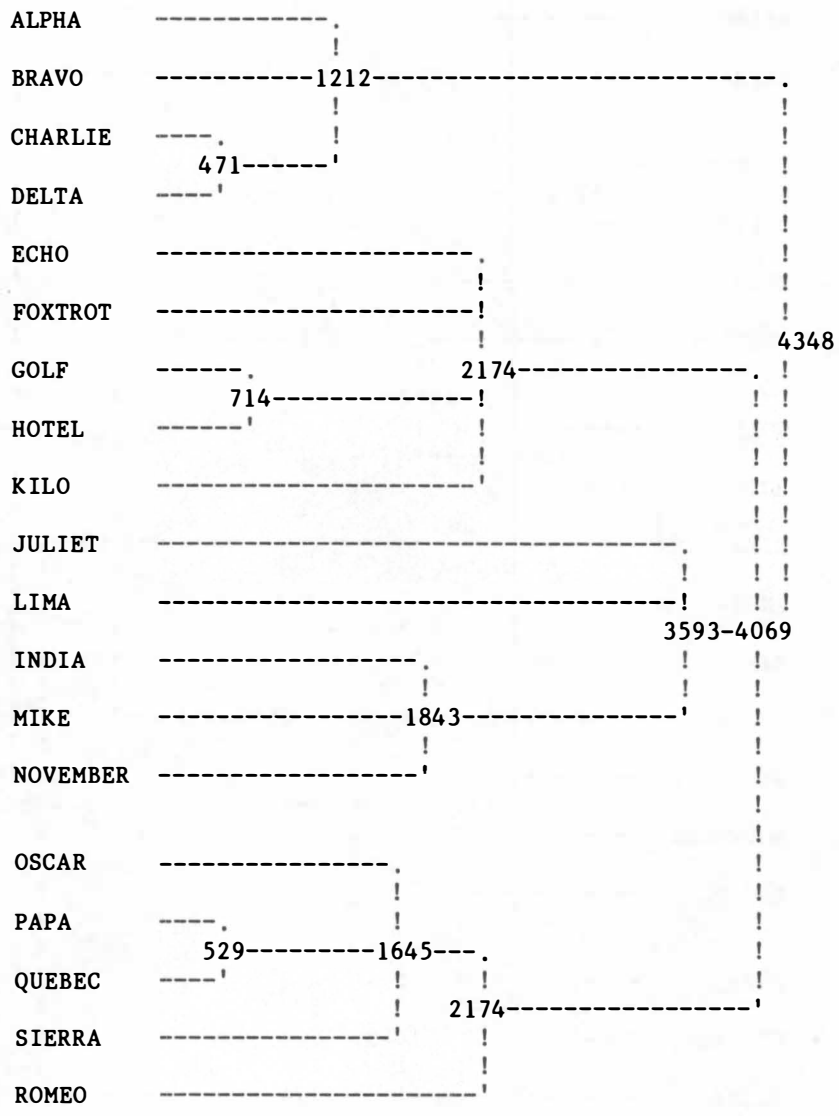
[510]

Fifth experiment: cognate percentages



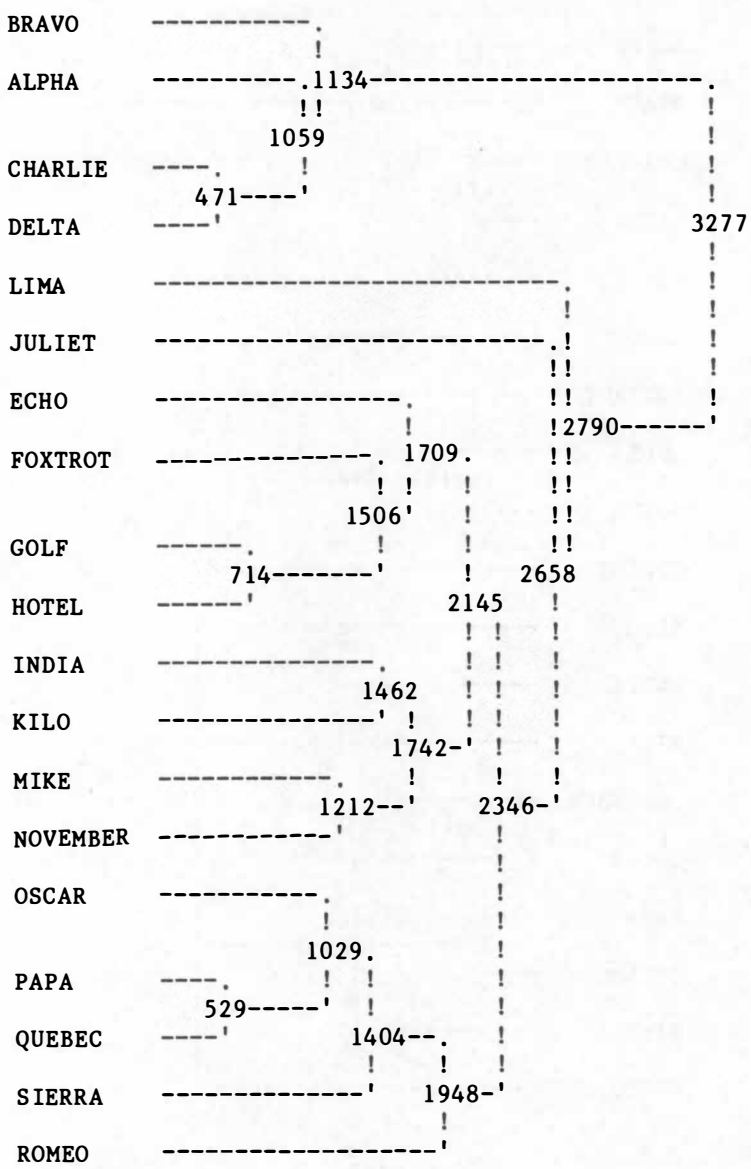
[521]

Fifth experiment: traditional lexicostatistical techniques, minimum-percentage method, no tolerance



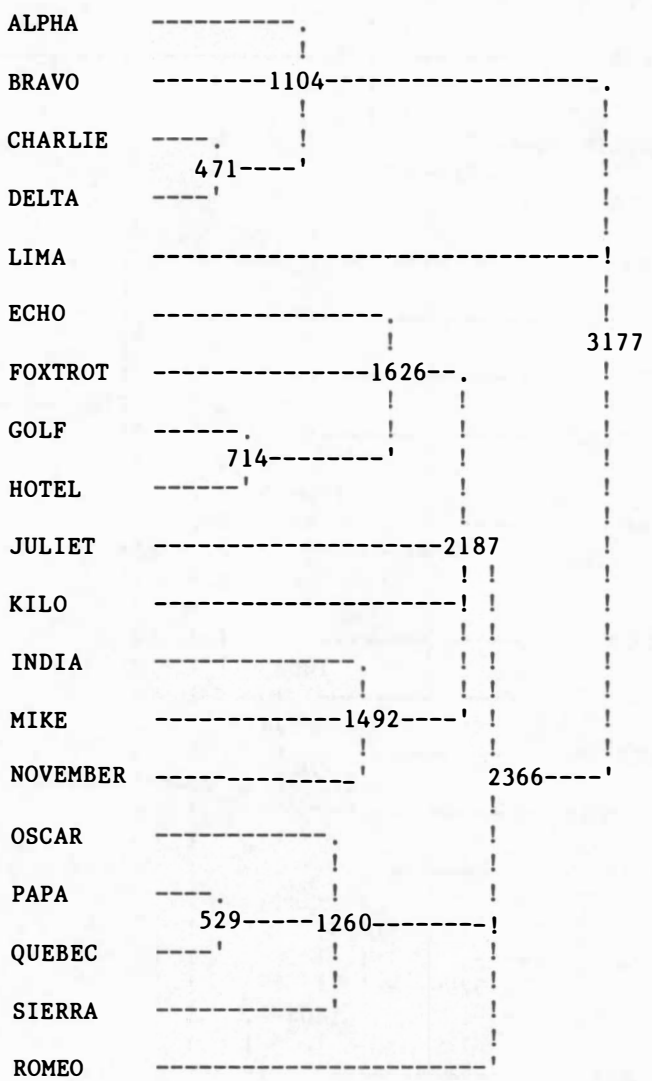
[522]

Fifth experiment: traditional lexicostatistical techniques, minimum-percentage method, 0.5 confidence level



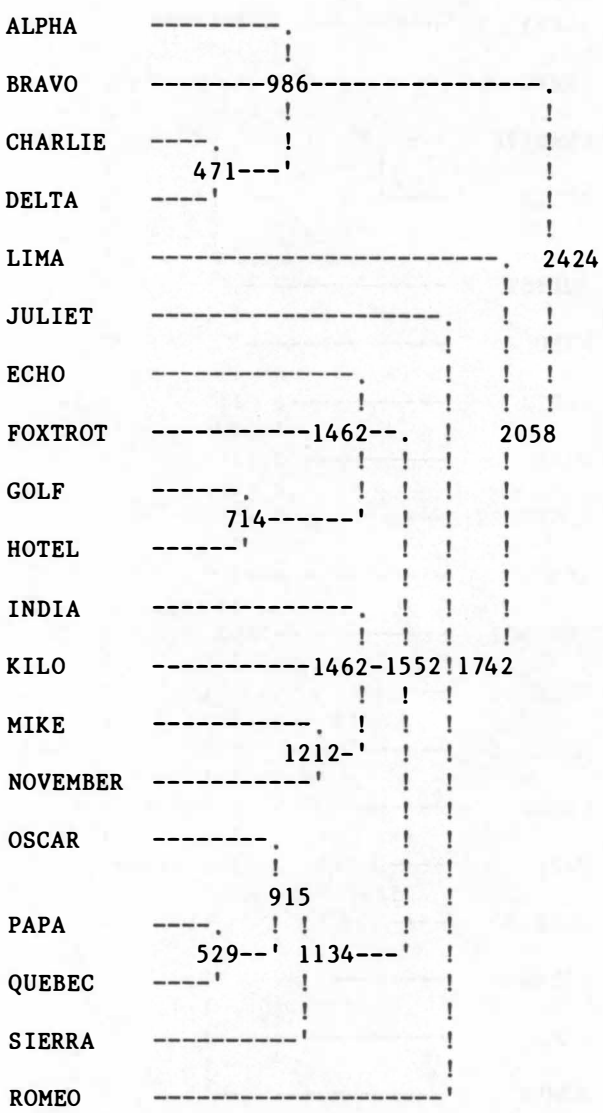
[523]

Fifth experiment: traditional lexicostatistical techniques, mean-percentage method, no tolerance



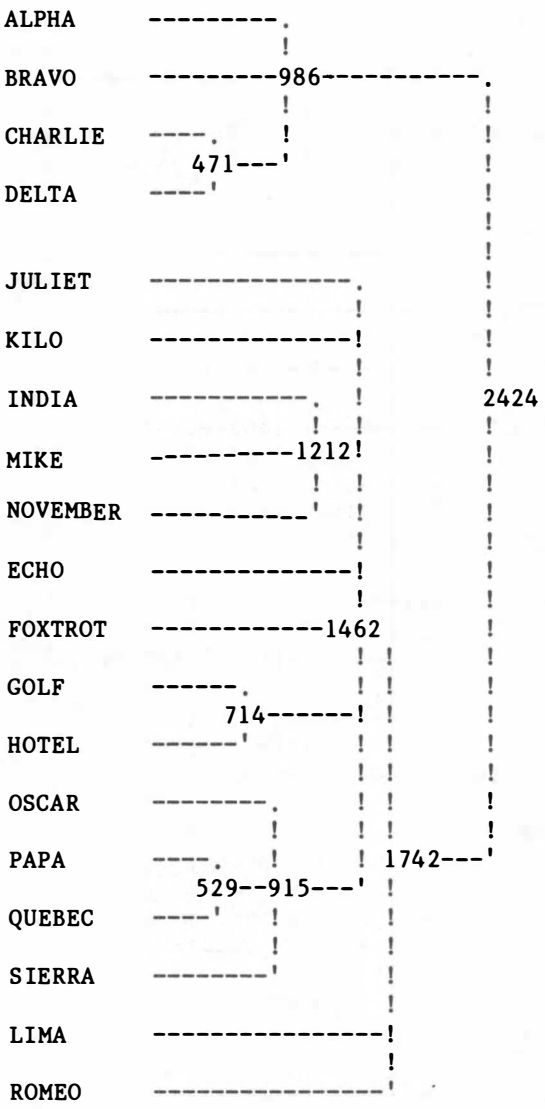
[524]

Fifth experiment: traditional lexicostatistical techniques, mean-percentage method, 0.5 confidence level



[525]

Fifth experiment: traditional lexicostatistical techniques, maximum-percentage method, no tolerance



[526]

Fifth experiment: traditional lexicostatistical techniques, maximum-percentage method, 0.5 confidence level

ALPHA

BRAVO	987	BRAVO									
CHARLIE	974	985	CHARLIE								
DELTA	980	974	992	DELTA							
ECHO	-416	-471	-448	-469	ECHO						
FOXTROT	-114	-201	-185	-212	819	FOXTROT					
GOLF	-489	-489	-459	-474	948	805	GOLF				
HOTEL	-493	-497	-488	-506	942	796	955	HOTEL			
INDIA	-418	-437	-444	-444	695	685	617	645	INDIA		
JULIET	-448	-453	-423	-443	669	605	653	585	932	JULIET	
KILO	-524	-521	-502	-544	782	723	722	711	942	916	KILO
LIMA	-257	-222	-218	-262	495	547	485	432	719	778	LIMA
MIKE	-163	-151	-125	-183	277	364	240	202	637	602	MIKE
NOVEMBER	-157	-229	-201	-204	467	411	403	302	664	761	NOVEMBER
OSCAR	-492	-518	-529	-521	580	370	489	577	371	456	OSCAR
PAPA	-465	-472	-482	-499	512	182	461	466	184	349	PAPA
QUEBEC	-451	-459	-488	-505	519	269	504	520	377	441	QUEBEC
ROMEO	-488	-487	-504	-541	655	347	539	650	428	516	ROMEO
SIERRA	-633	-622	-630	-633	677	390	678	696	438	505	SIERRA

ALP BRA CHA DEL ECH FOX GOL HOT IND JUL

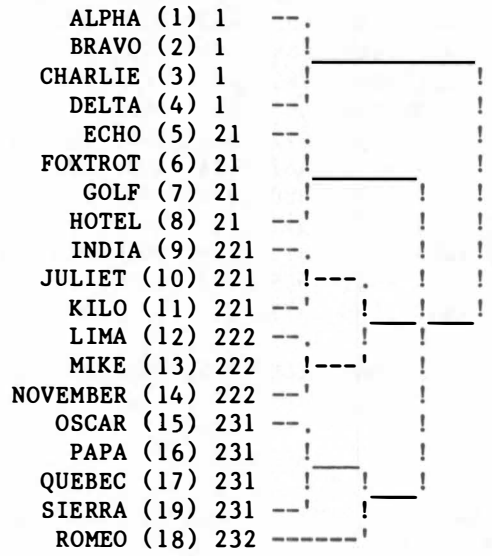
KILO

LIMA	736	LIMA						
MIKE	645	749	MIKE					
NOVEMBER	689	863	897	NOVEMBER				
OSCAR	487	443	51	193	OSCAR			
PAPA	359	282	17	71	956	PAPA		
QUEBEC	396	351	50	136	953	972	QUEBEC	
ROMEO	512	447	61	237	917	888	896	ROMEO
SIERRA	490	349	102	119	918	834	882	922	SIERRA

KIL LIM MIK NOV OSC PAP QUE ROM

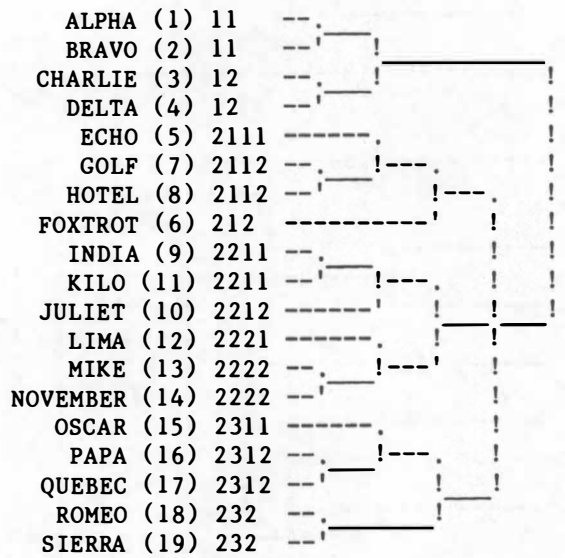
[530]

Fifth experiment: correlation coefficients



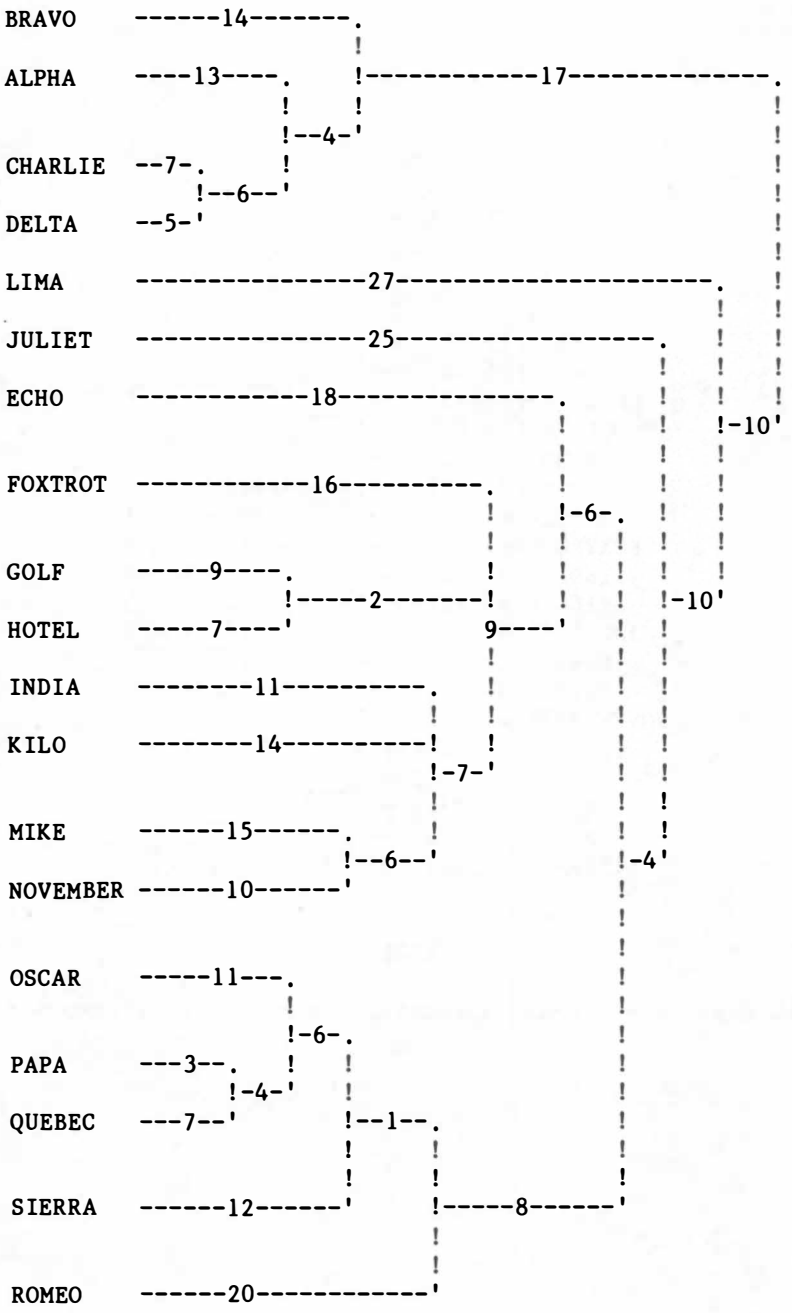
[547]

Fifth experiment: n-way splitting algorithm, with recomputations



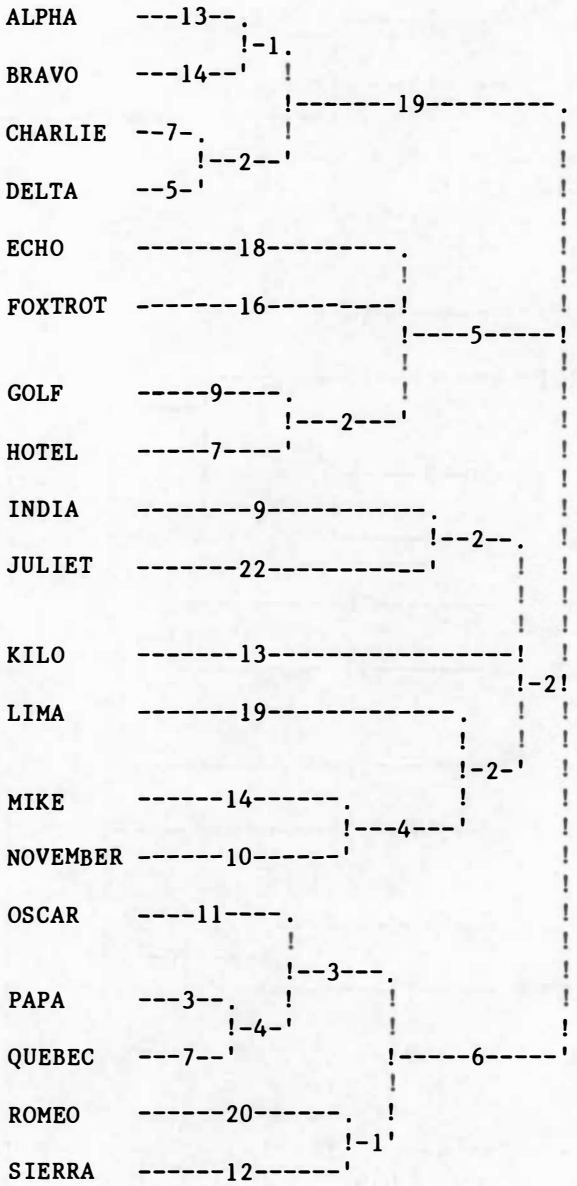
[548]

Fifth experiment: n-way splitting algorithm, no recomputations



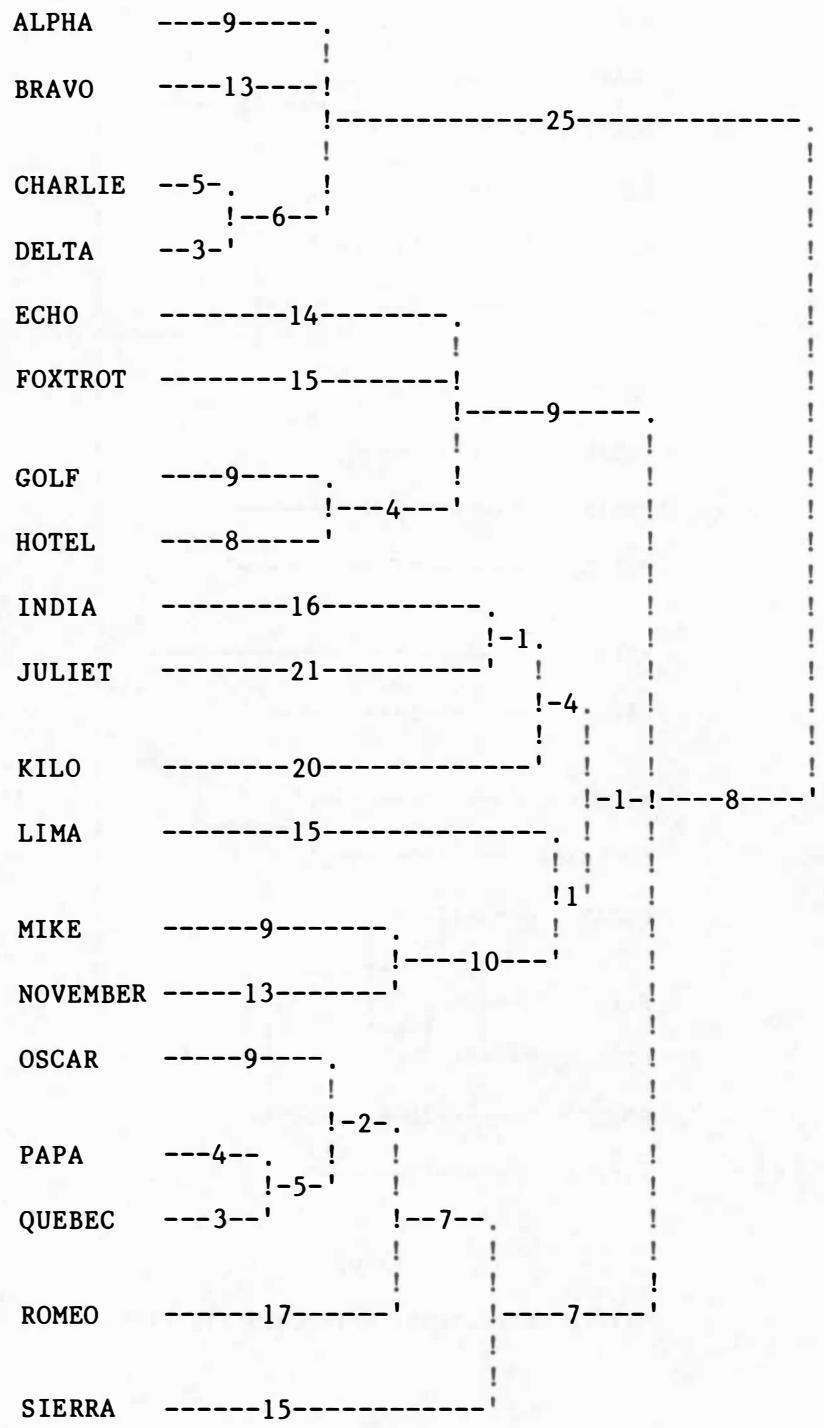
[550]

Fifth experiment: reduced mutation algorithm



[560]

Fifth experiment: wild-card algorithm



[600]

Sixth experiment: 50-item wordlists, 4000-year time depth

ALPHA

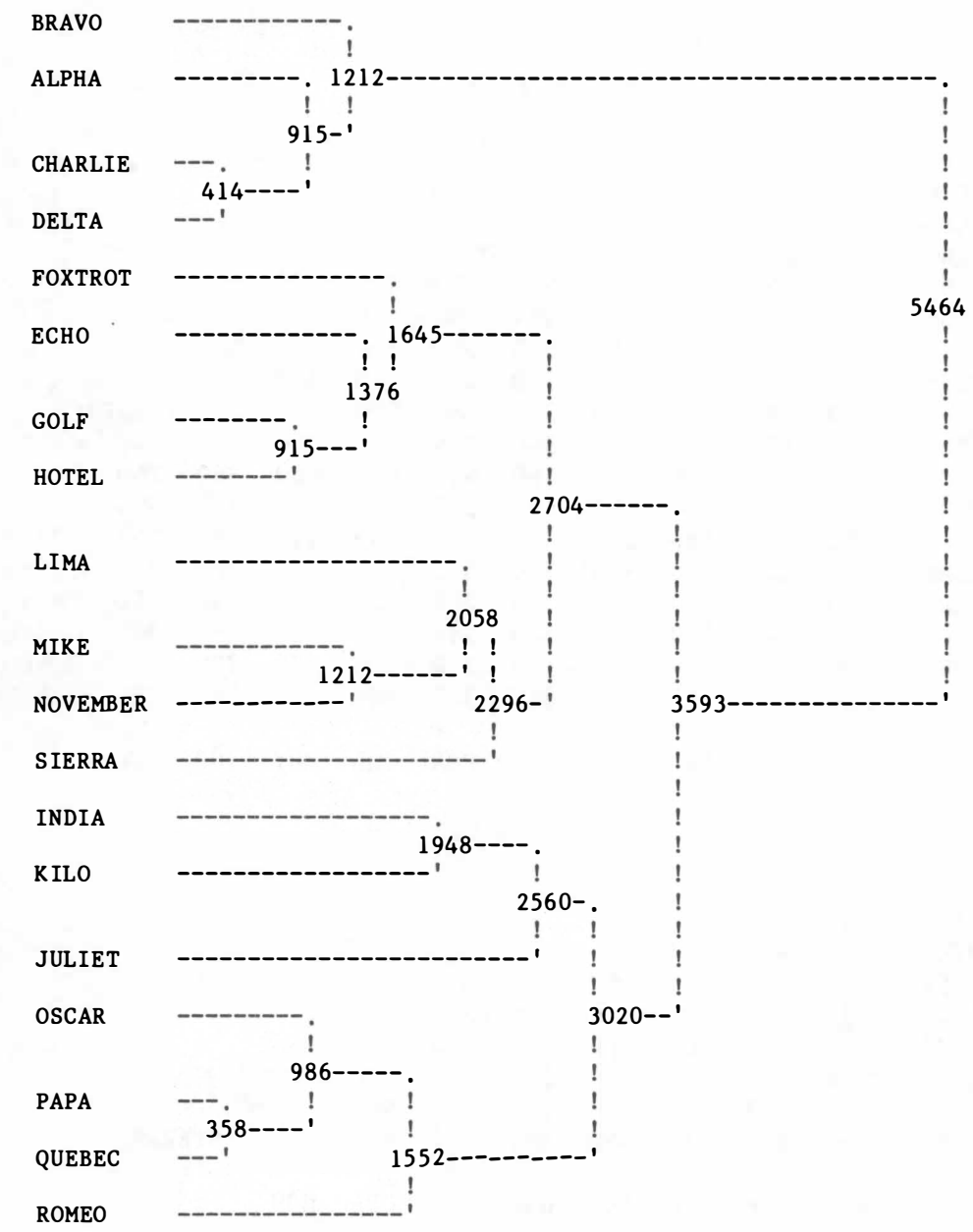
BRAVO	600	BRAVO									
CHARLIE	680	600	CHARLIE								
DELTA	700	620	840	DELTA							
ECHO	220	200	220	220	ECHO						
FOXTROT	200	160	140	160	520	FOXTROT					
GOLF	180	180	180	160	580	500	GOLF				
HOTEL	200	200	160	160	560	560	680	HOTEL			
INDIA	240	220	240	220	340	240	340	340	INDIA		
JULIET	200	180	200	180	360	280	320	320	400	JULIET	
KILO	200	180	180	200	340	280	280	300	440	340	KILO
LIMA	220	240	200	220	460	400	440	400	340	360	LIMA
MIKE	200	200	180	160	380	360	380	360	360	300	MIKE
NOVEMBER	220	200	180	180	320	320	340	360	280	240	NOVEMBER
OSCAR	140	120	100	120	380	340	360	400	300	300	OSCAR
PAPA	140	140	140	120	380	280	380	420	380	340	PAPA
QUEBEC	140	140	160	140	380	280	400	420	380	340	QUEBEC
ROMEO	100	100	100	100	280	220	300	320	320	300	ROMEO
SIERRA	220	200	160	160	360	340	400	480	340	260	SIERRA
	ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL	

KILO

LIMA	300	LIMA						
MIKE	280	480	MIKE					
NOVEMBER	240	420	600	NOVEMBER				
OSCAR	280	380	400	400	OSCAR			
PAPA	320	300	400	380	660	PAPA		
QUEBEC	320	360	460	420	680	860	QUEBEC	
ROMEO	280	220	280	240	520	560	560	ROMEO
SIERRA	300	380	380	420	480	480	460	340
	KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

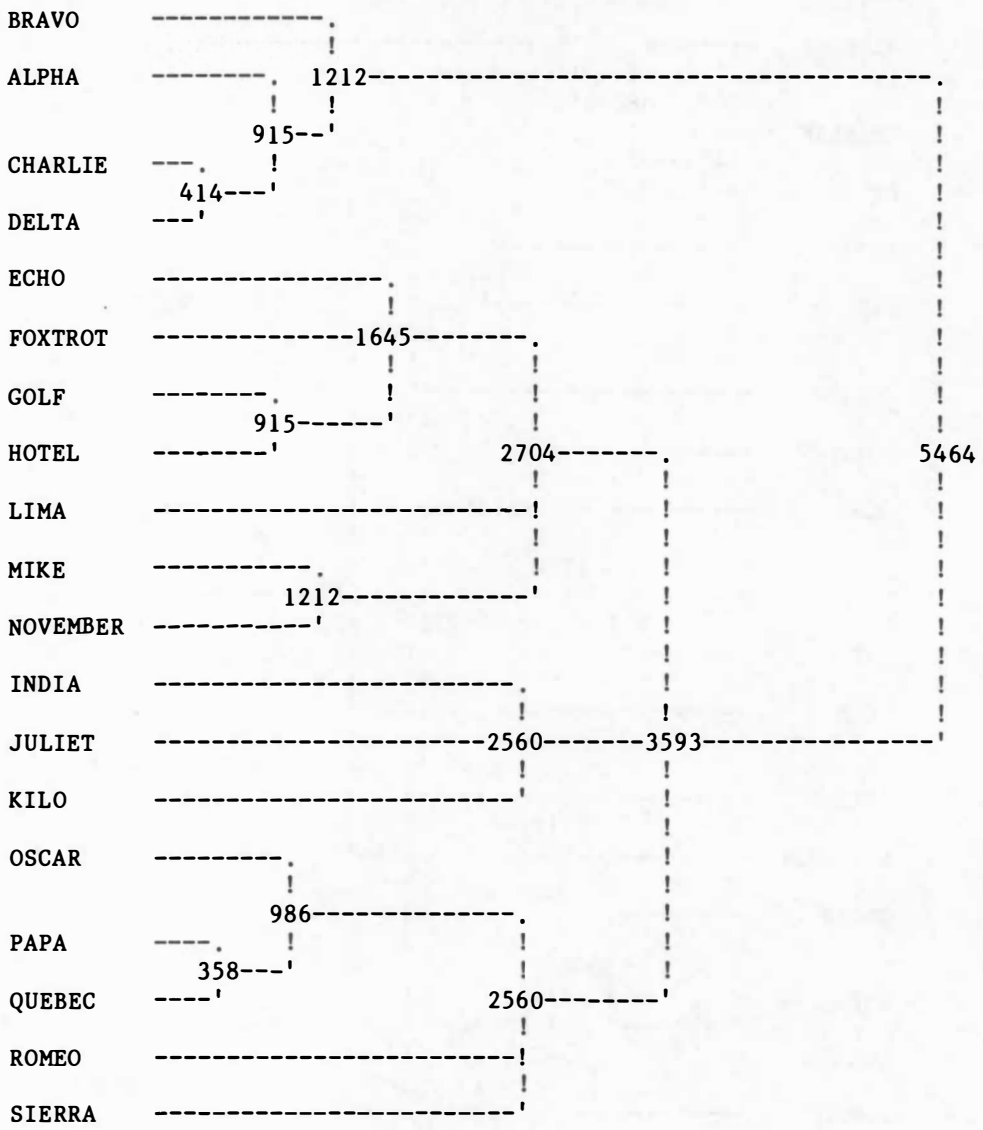
[610]

Sixth experiment: cognate percentages



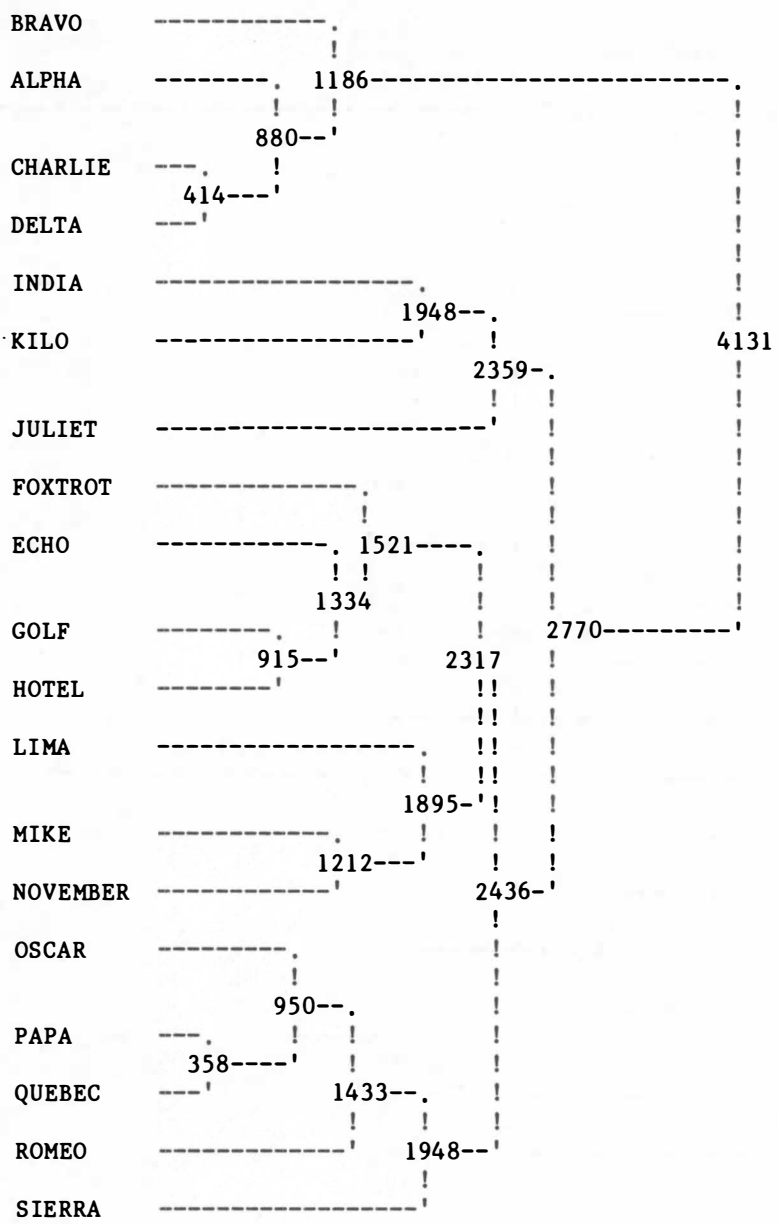
[621]

Sixth experiment: traditional lexicostatistical techniques, minimum-percentage method, no tolerance



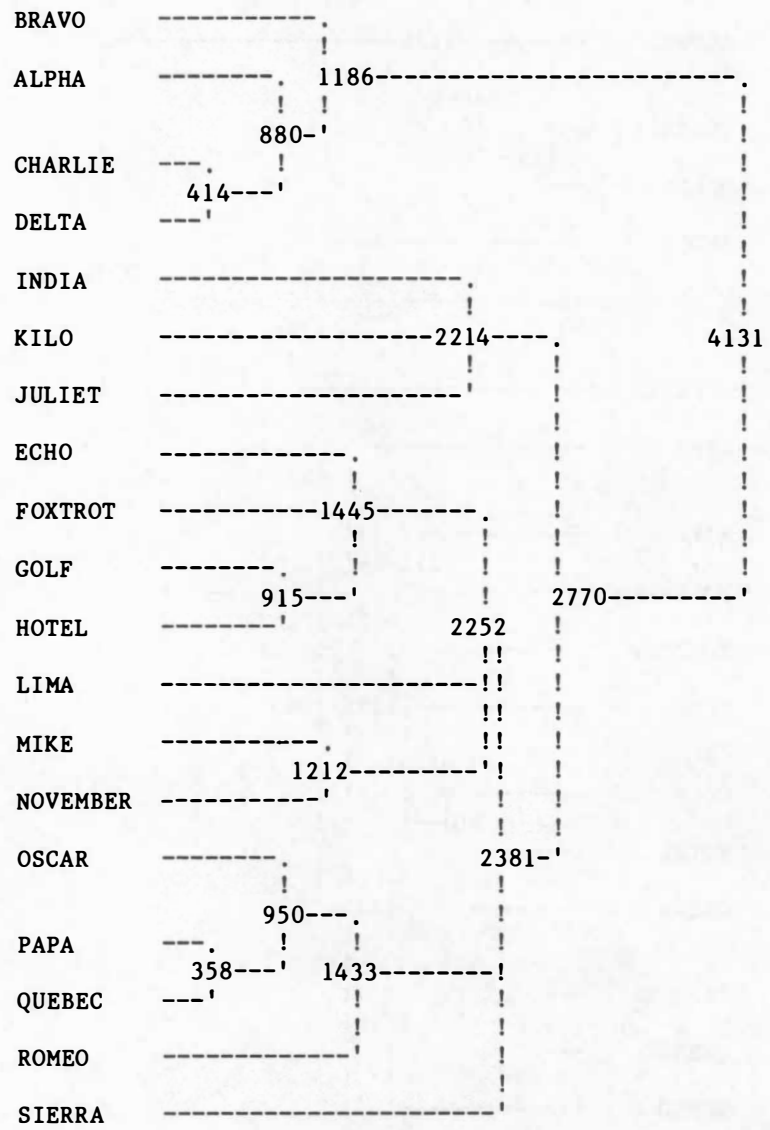
[622]

Sixth experiment: traditional lexicostatistical techniques, minimum-percentage method, 0.5 confidence level



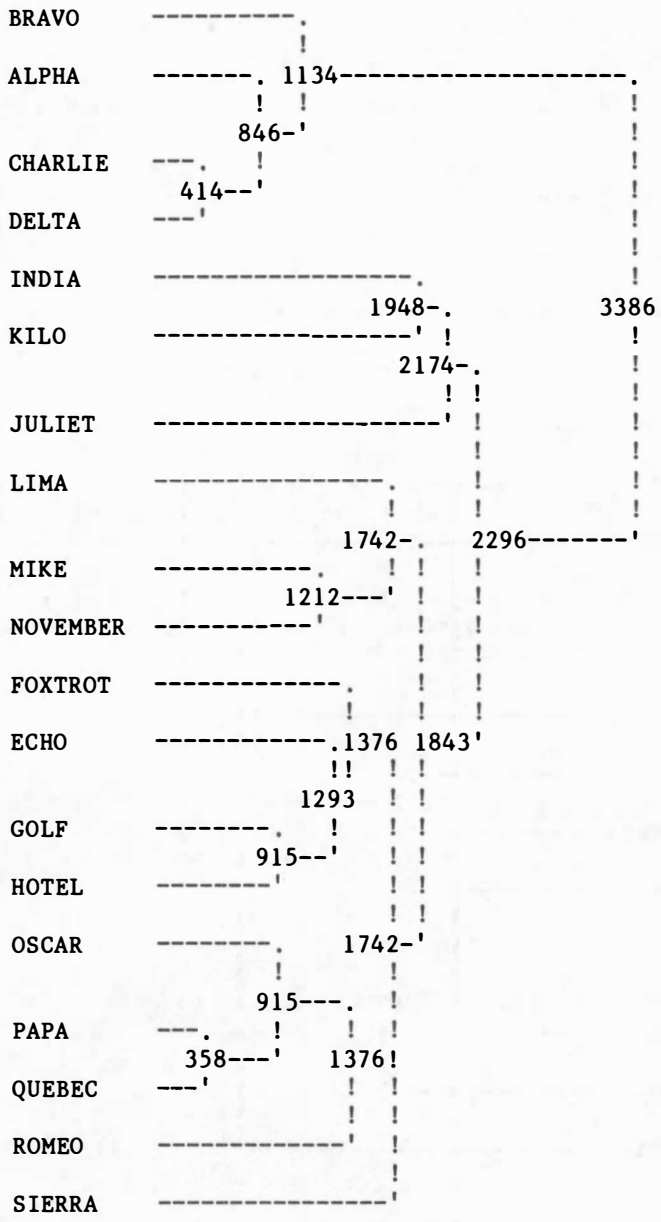
[623]

Sixth experiment: traditional lexicostatistical techniques, mean-percentage method, no tolerance



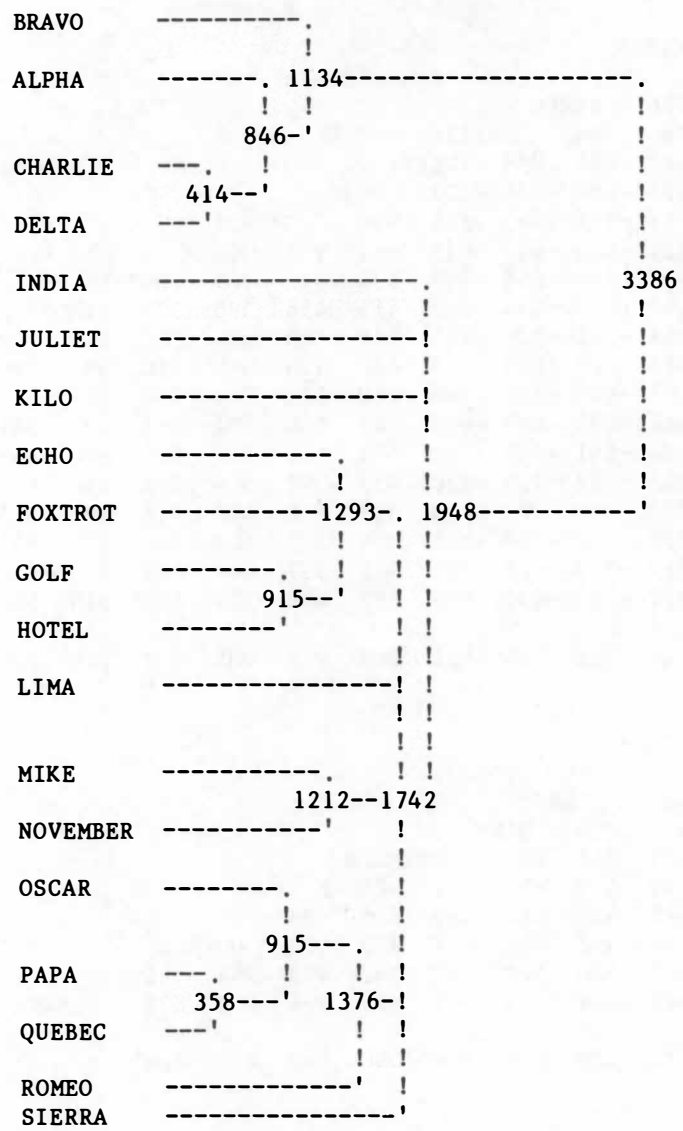
[624]

Sixth experiment: traditional lexicostatistical techniques, mean-percentage method, 0.5 confidence level



[625]

Sixth simulation: traditional lexicostatistical technique, maximum-percentage method, no tolerance



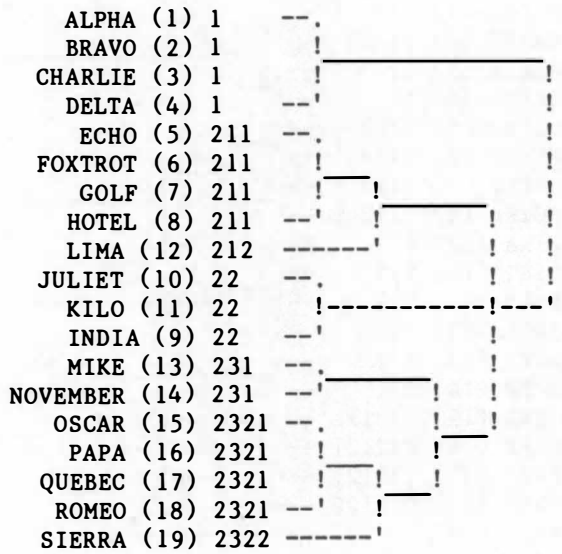
[626]

Sixth experiment: traditional lexicostatistical techniques, maximum-percentage method, 0.5 confidence level

```
          ALPHA

  BRAVO   996   BRAVO
CHARLIE   984  984   CHARLIE
  DELTA   987  984  995   DELTA
   ECHO  -624 -604 -636 -637   ECHO
FOXTROT  -566 -521 -543 -558  956   FOXTROT
   GOLF  -622 -614 -643 -625  961  957   GOLF
  HOTEL  -675 -676 -665 -669  938  912  969   HOTEL
  INDIA  -697 -663 -654 -642  415  456  395  375   INDIA
 JULIET  -768 -741 -726 -713  619  566  636  589  843   JULIET
   KILO  -683 -670 -623 -679  480  420  565  510  885  926  KILO
   LIMA  -607 -620 -615 -648  768  827  762  780  417  535  LIMA
   MIKE  -689 -673 -693 -679  547  554  582  611  391  549  MIKE
NOVEMBER -584 -541 -569 -580  541  548  553  525  446  485  NOVEMBER
  OSCAR  -805 -787 -753 -780  455  391  544  565  616  616  OSCAR
   PAPA  -729 -728 -687 -690  351  330  428  448  537  555  PAPA
 QUEBEC  -733 -728 -708 -711  366  347  405  446  537  563  QUEBEC
  ROMEO  -749 -749 -698 -717  425  337  441  485  638  640  ROMEO
 SIERRA  -827 -797 -801 -804  687  668  724  690  519  688  SIERRA

          ALP  BRA  CHA  DEL  ECH  FOX  GOL  HOT  IND  JUL


          KILO

   LIMA   504   LIMA
   MIKE   478  776   MIKE
NOVEMBER  382  762  937   NOVEMBER
  OSCAR   550  429  684  641   OSCAR
   PAPA   512  424  583  556  956   PAPA
 QUEBEC   496  352  561  567  957  989   QUEBEC
  ROMEO   605  430  572  572  941  976  964   ROMEO
 SIERRA   541  659  813  746  875  794  832  828   SIERRA

          KIL  LIM  MIK  NOV  OSC  PAP  QUE  ROM
```

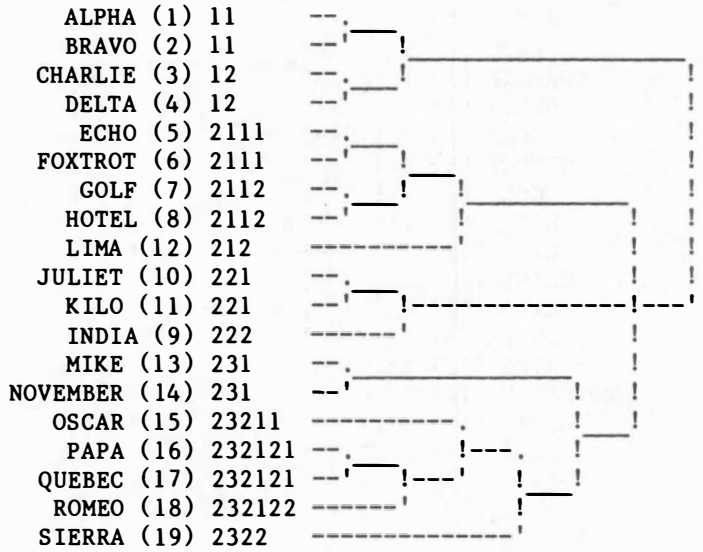
[630]

Sixth experiment: correlation coefficients



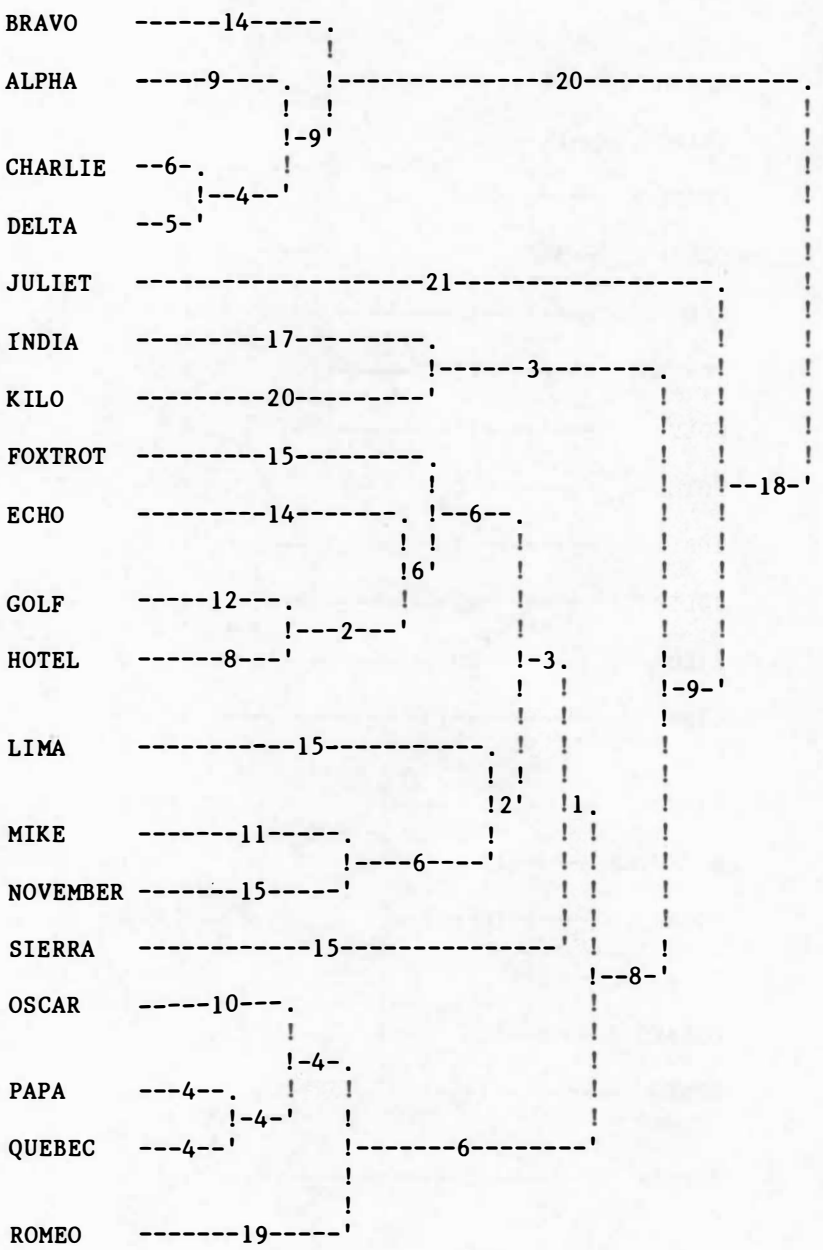
[647]

Sixth experiment: n-way splitting algorithm, with recomputations



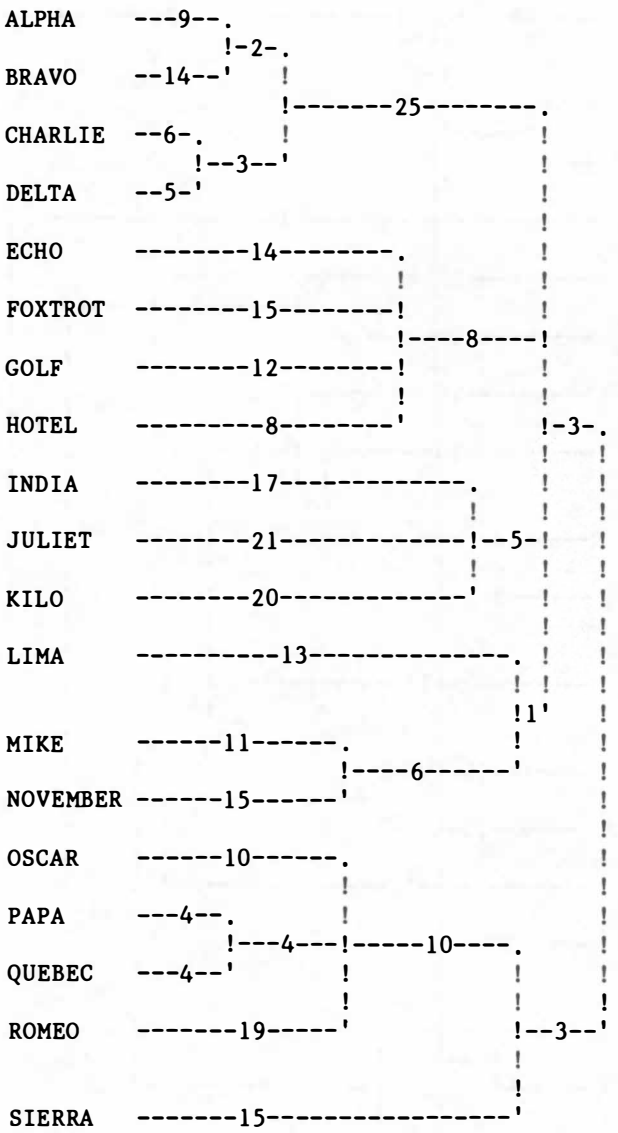
[648]

Sixth experiment: n-way splitting algorithm, no recomputations



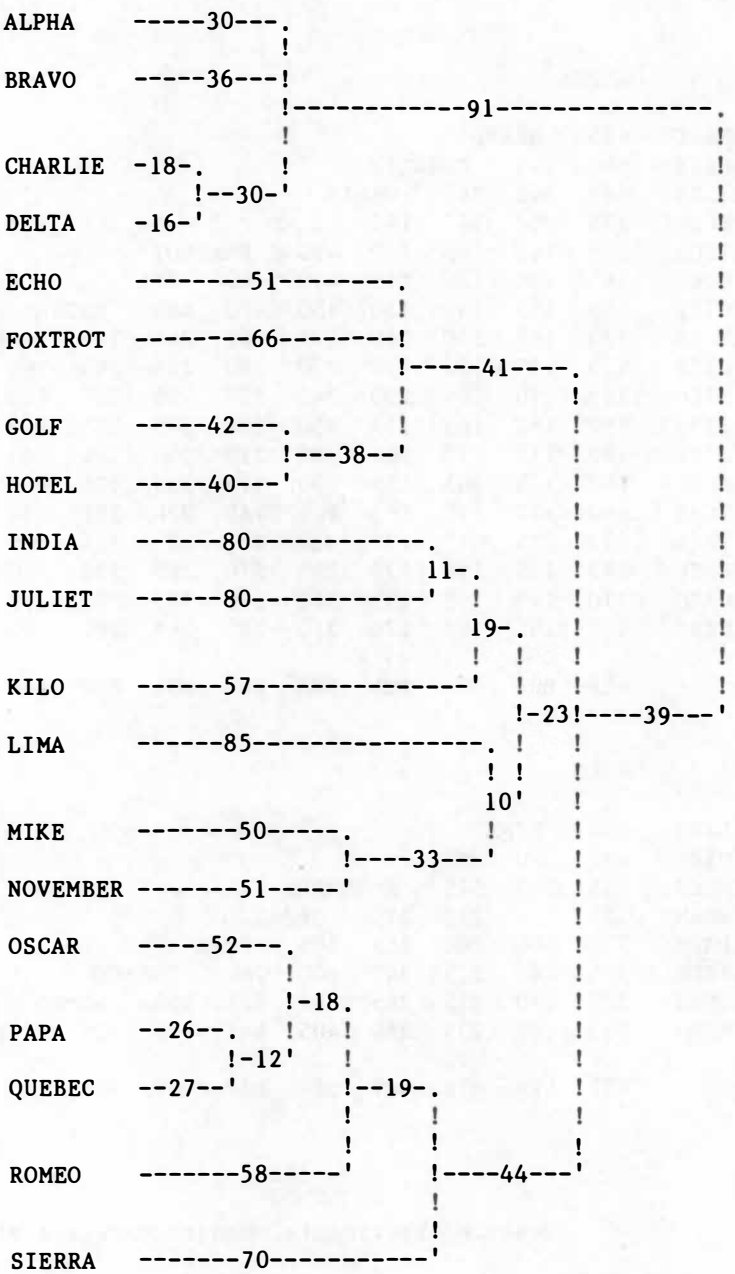
[650]

Sixth experiment: reduced mutation algorithm



[660]

Sixth experiment: wild-card algorithm



[700]

Seventh experiment: 200-item wordlists, 4000-year time depth

ALPHA

BRAVO	695									
CHARLIE	660	640								
DELTA	680	640	845							
ECHO	175	200	185	185						
FOXTROT	165	160	145	130	490					
GOLF	145	130	120	115	450	440				
HOTEL	155	155	140	130	480	470	645			
INDIA	130	145	140	140	275	260	240	245		
JULIET	150	175	150	150	270	250	235	245	360	
KILO	215	210	205	200	340	325	290	300	420	380
LIMA	160	180	165	155	250	260	255	255	295	305
MIKE	180	175	170	160	300	290	300	280	300	325
NOVEMBER	180	175	165	155	290	310	305	305	305	310
OSCAR	190	170	170	165	265	265	270	250	230	245
PAPA	215	225	215	195	310	290	285	255	245	240
QUEBEC	185	205	190	170	295	270	260	230	240	225
ROMEO	170	175	175	155	310	270	285	270	260	245
SIERRA	175	185	160	170	310	265	265	280	260	230

ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL

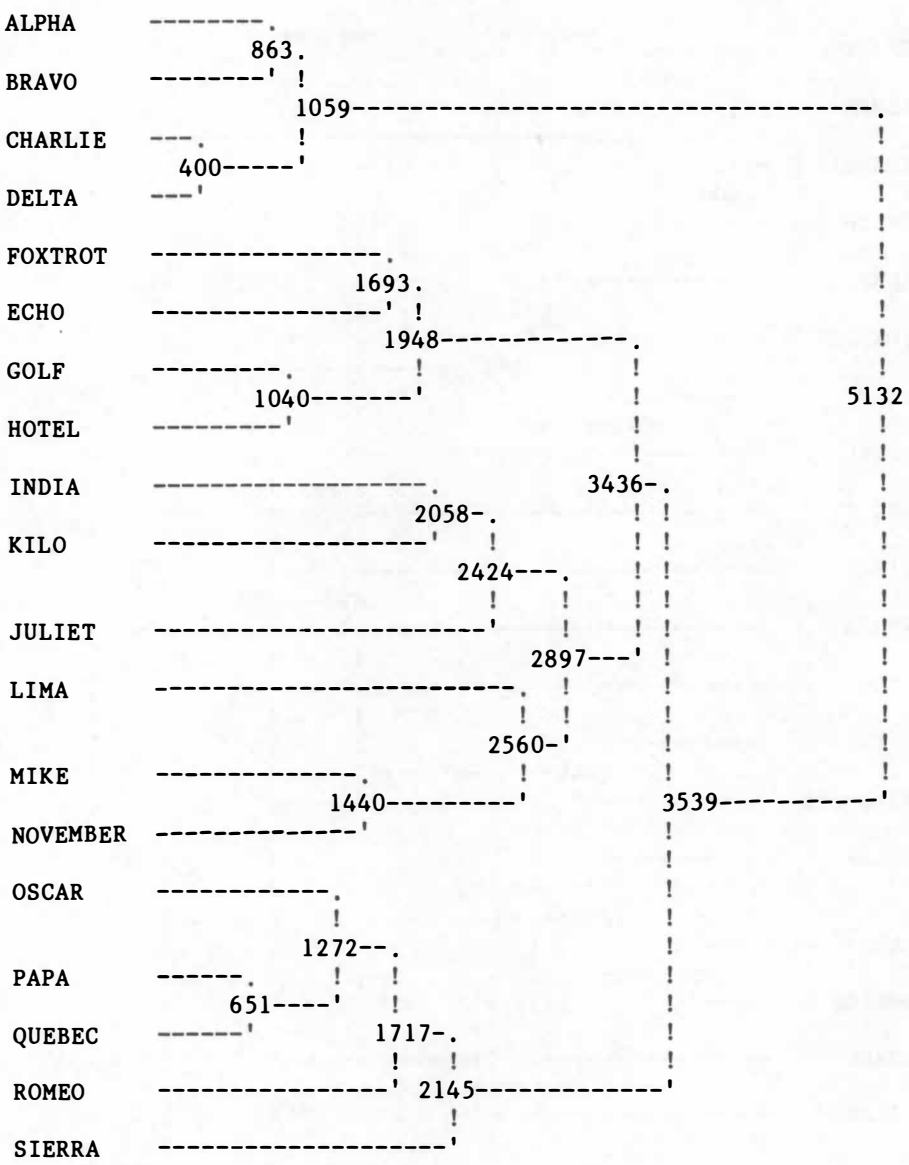
KILO

LIMA	350						
MIKE	420	340					
NOVEMBER	415	360	545				
OSCAR	305	255	270	270			
PAPA	335	260	280	265	585		
QUEBEC	315	245	275	245	600	760	
ROMEO	320	270	275	265	485	525	530
SIERRA	315	260	235	250	405	445	425	405

KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

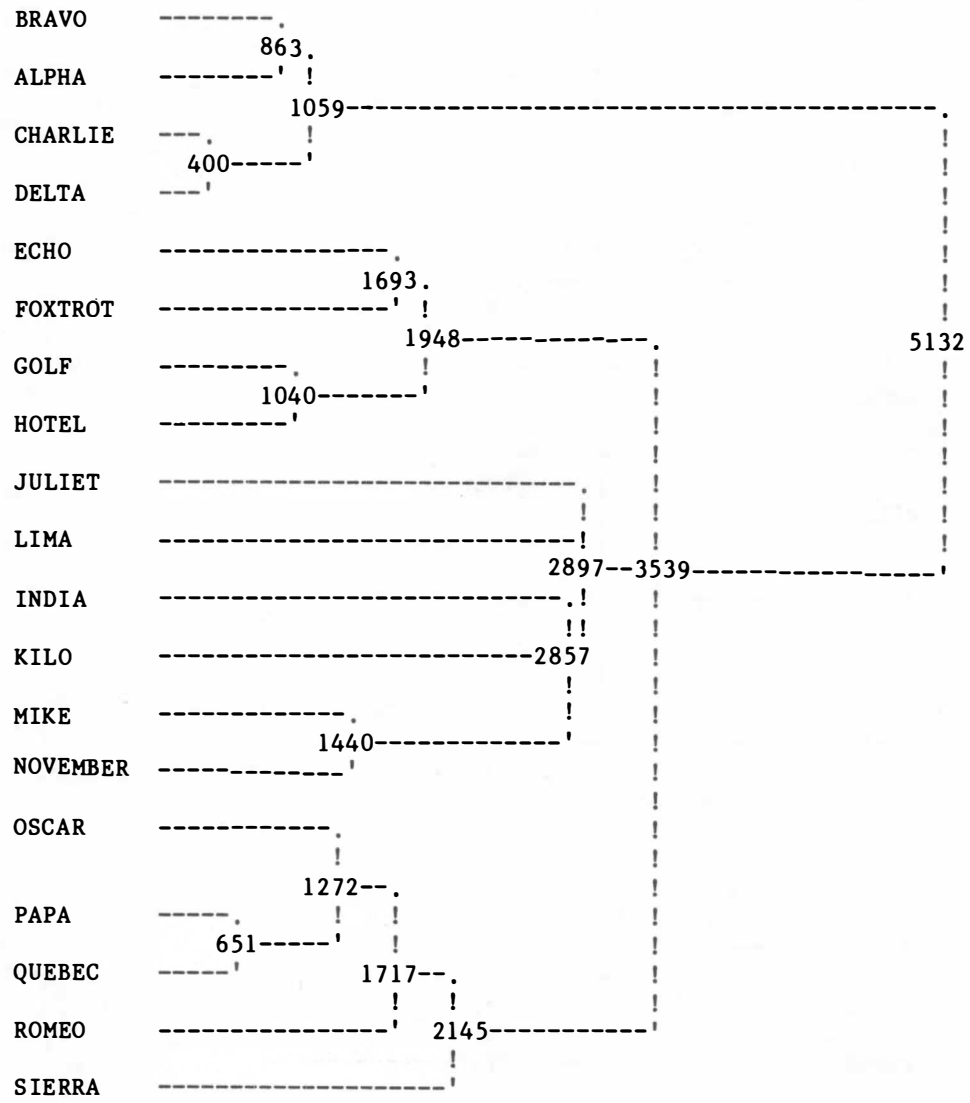
[710]

Seventh experiment: cognate percentages



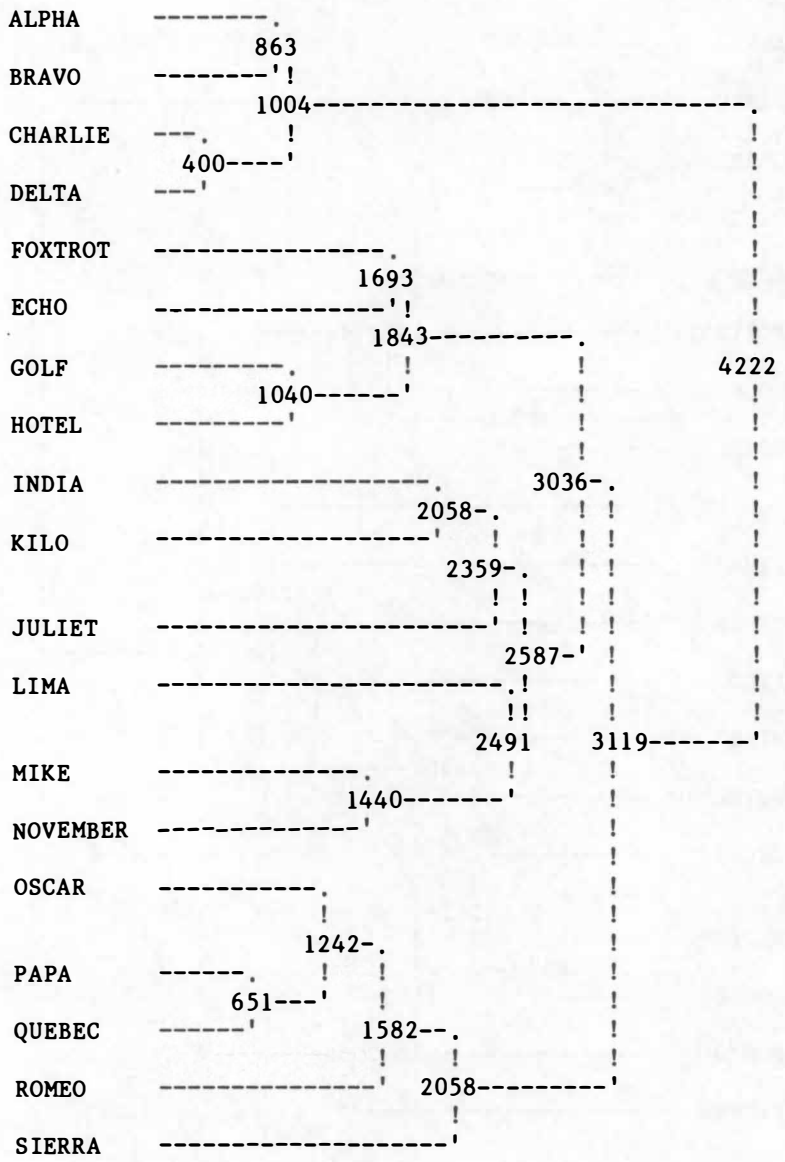
[721]

Seventh experiment: traditional lexicostatistical techniques, minimum-percentage method, no tolerance



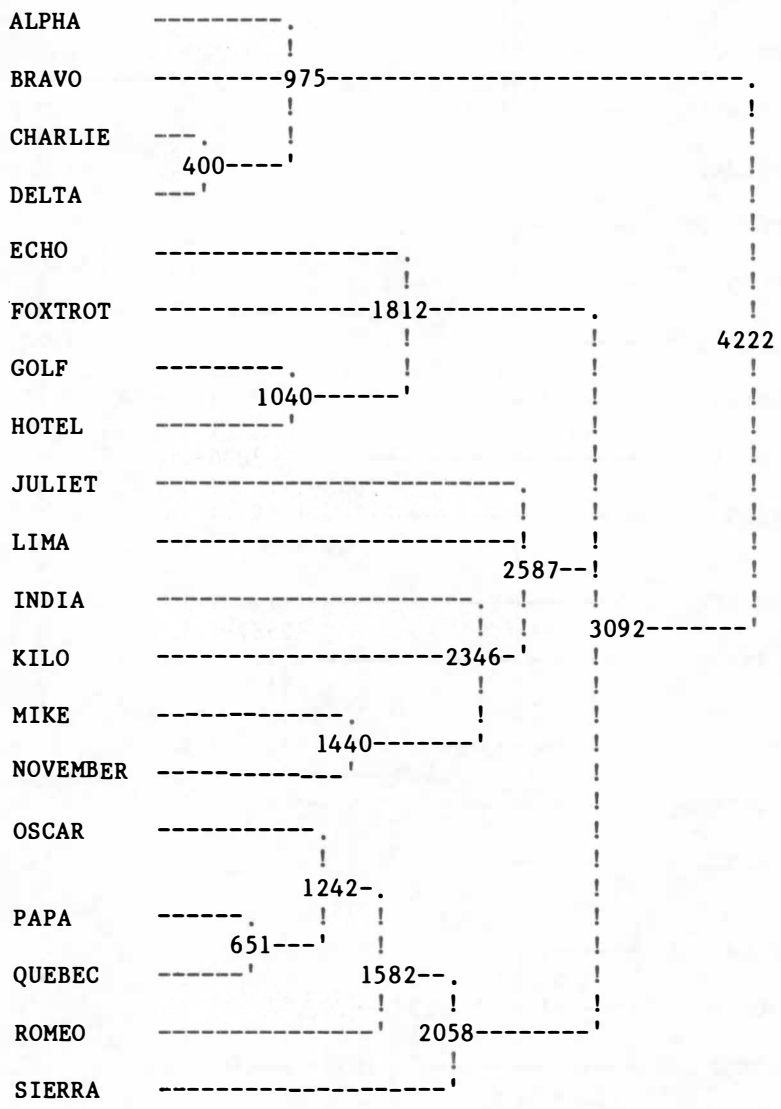
[722]

Seventh experiment: traditional lexicostatistical techniques, minimum-percentage method, 0.5 confidence level



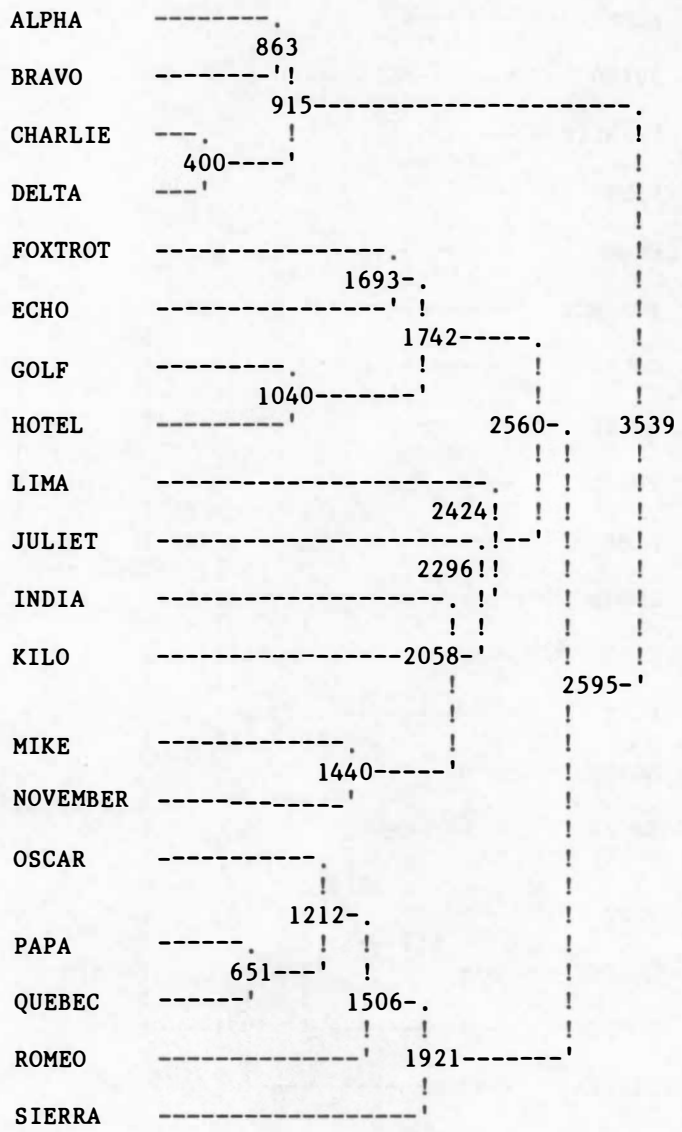
[723]

Seventh experiment: traditional lexicostatistical techniques, mean-percentage method, no tolerance



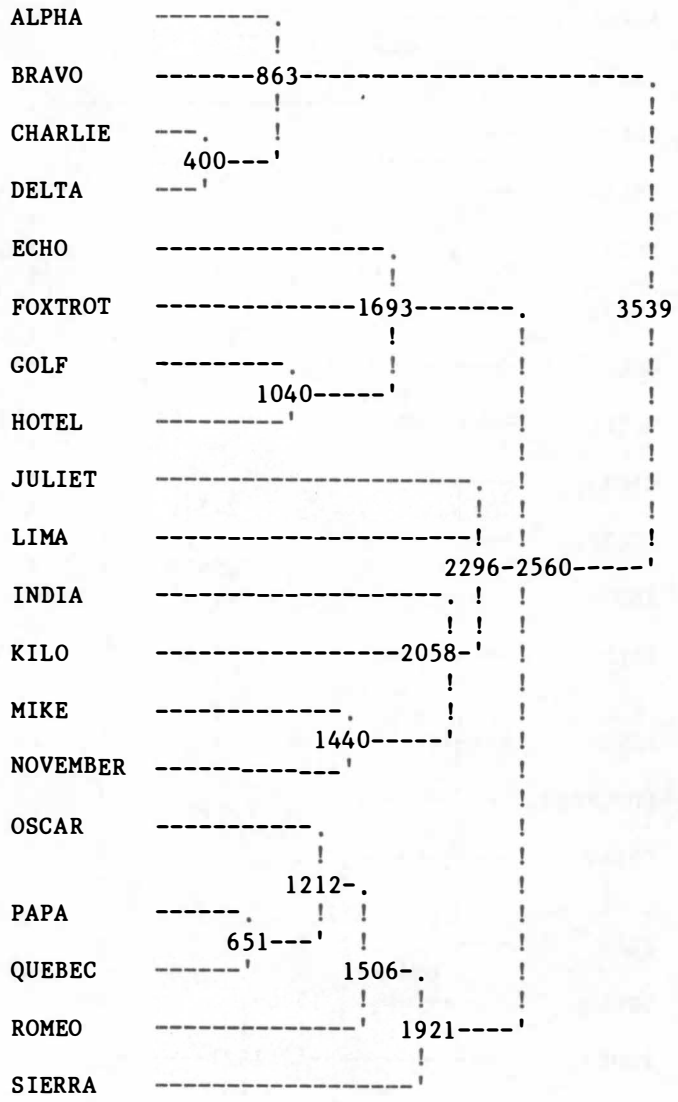
[724]

Seventh experiment: traditional lexicostatistical techniques, mean-percentage method, 0.5 confidence level



[725]

Seventh experiment: traditional lexicostatistical techniques, maximum-percentage method, no tolerance



[726]

Seventh experiment: traditional lexicostatistical techniques, maximum-percentage method, 0.5 confidence level

ALPHA

BRAVO	996	BRAVO									
CHARLIE	978	974	CHARLIE								
DELTA	973	978	999	DELTA							
ECHO	-618	-662	-639	-645	ECHO						
FOXTROT	-675	-675	-670	-652	978	FOXTROT					
GOLF	-620	-616	-610	-609	932	944	GOLF				
HOTEL	-563	-580	-568	-555	914	920	979	HOTEL			
INDIA	-735	-739	-726	-723	426	495	415	378	INDIA		
JULIET	-701	-726	-692	-689	360	469	388	331	973	JULIET	
KILO	-806	-789	-782	-775	333	439	403	326	940	966	KILO
LIMA	-763	-787	-769	-756	437	483	416	368	919	927	LIMA
MIKE	-616	-616	-610	-601	354	473	373	380	796	799	MIKE
NOVEMBER	-627	-631	-619	-608	428	464	421	376	773	831	NOVEMBER
OSCAR	-437	-404	-411	-424	225	190	162	84	254	133	OSCAR
PAPA	-344	-349	-343	-337	111	104	88	32	141	55	PAPA
QUEBEC	-328	-347	-336	-331	127	132	117	57	160	112	QUEBEC
ROMEO	-518	-517	-521	-511	256	301	246	167	303	237	ROMEO
SIERRA	-566	-572	-544	-574	327	384	346	215	352	304	SIERRA

ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL

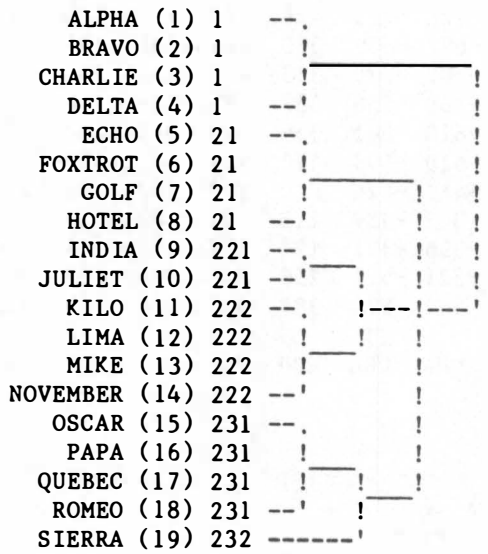
KILO

LIMA	959	LIMA					
MIKE	829	921	MIKE				
NOVEMBER	822	889	979	NOVEMBER			
OSCAR	267	263	156	85	OSCAR		
PAPA	117	146	51	4	979	PAPA	
QUEBEC	187	202	65	58	978	998	QUEBEC
ROMEO	322	318	206	169	983	950	951	ROMEO
SIERRA	338	354	270	183	948	901	921	983	SIERRA

KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

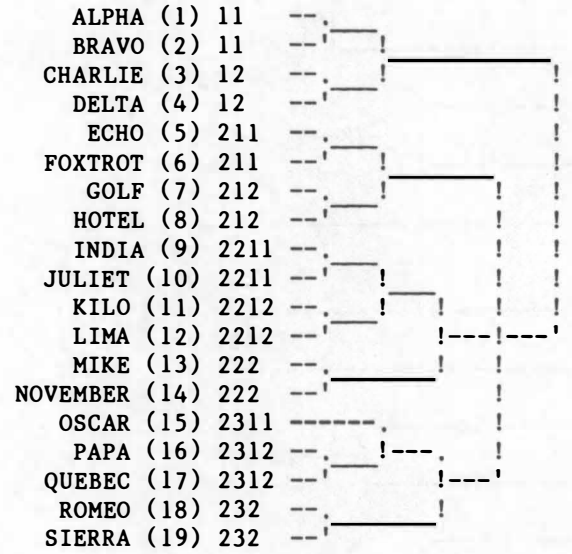
[730]

Seventh experiment: correlation coefficients



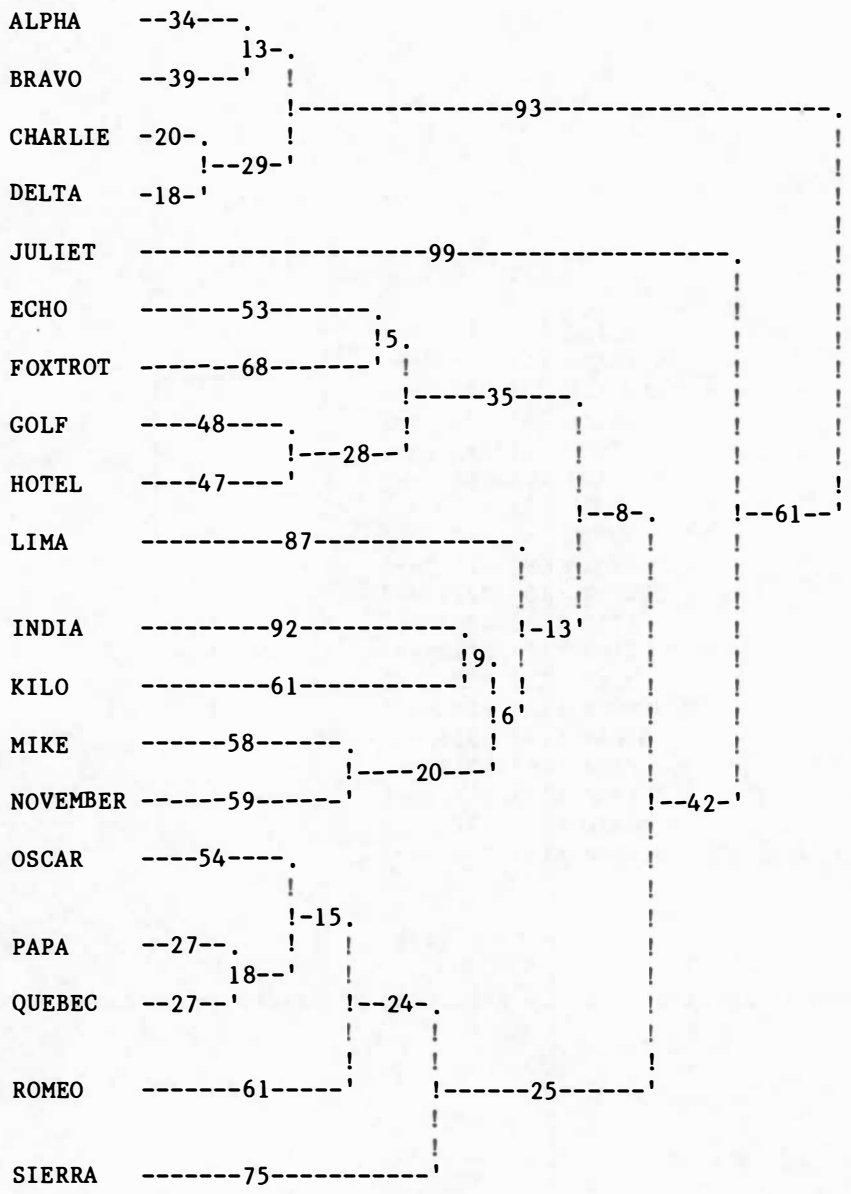
[747]

Seventh experiment: n-way splitting algorithm, with recomputations



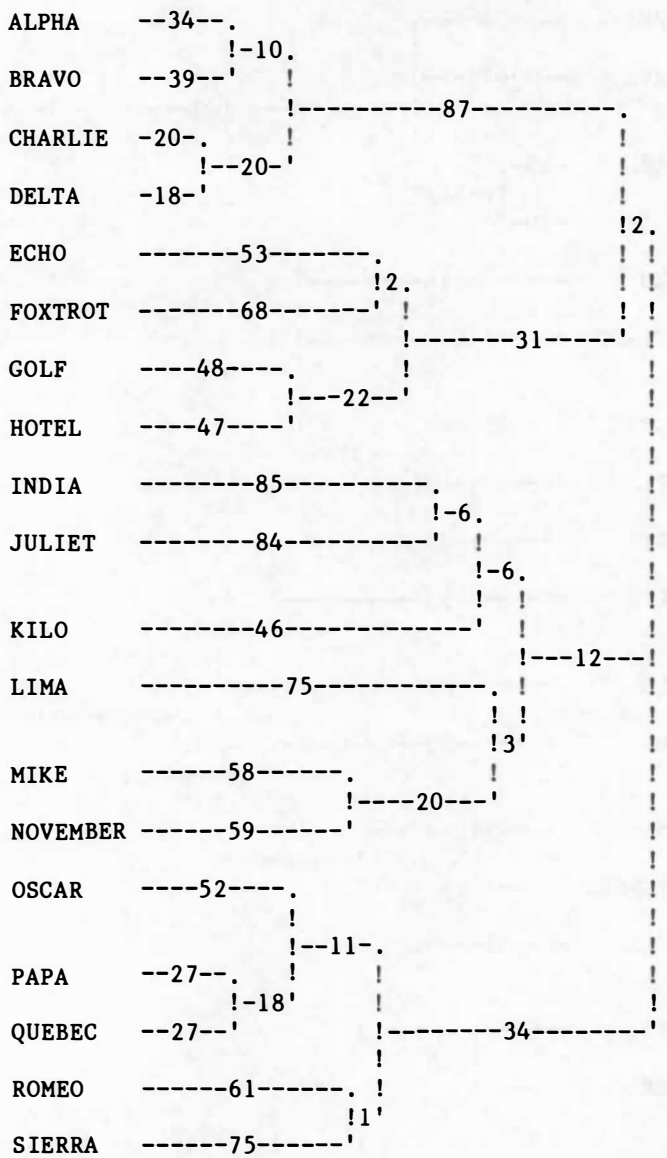
[748]

Seventh experiment: n-way splitting algorithm, no recomputations



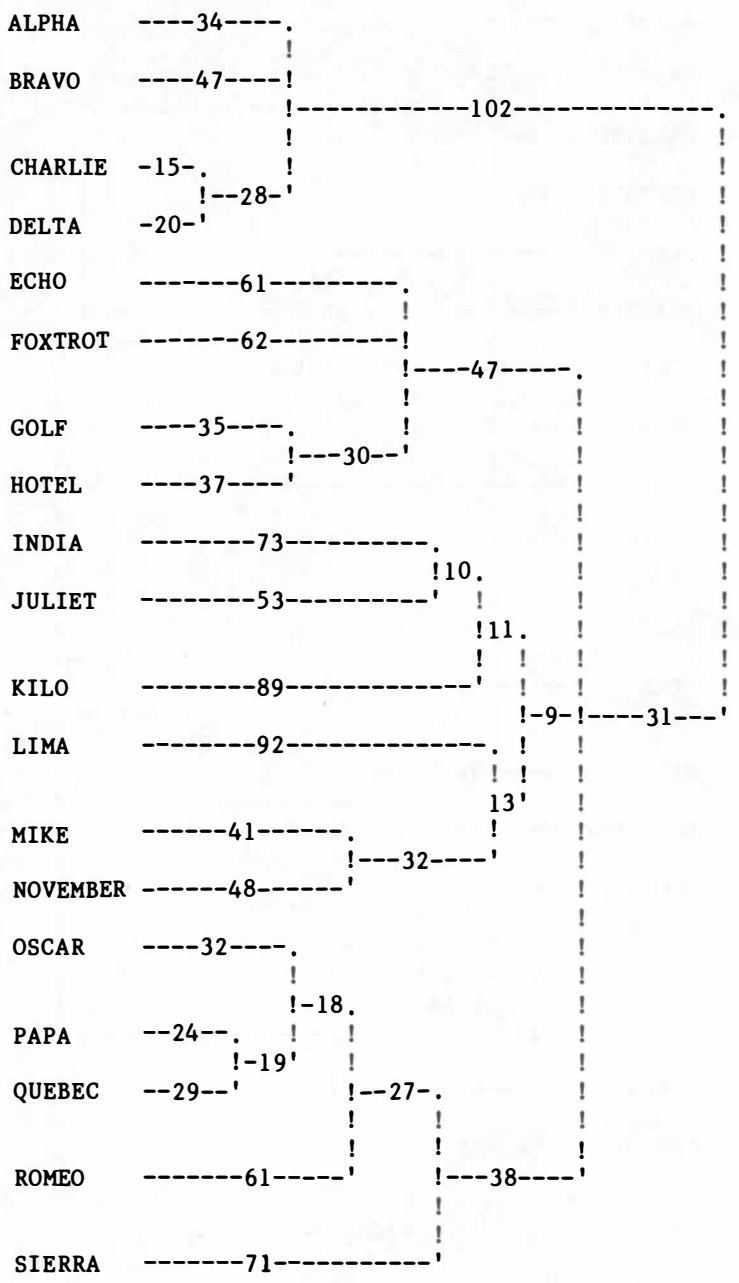
[750]

Seventh experiment: reduced mutation method



[760]

Seventh experiment: wild-card algorithm



[800]

Eighth experiment: 200-item wordlists, 4000-year time depth

ALPHA

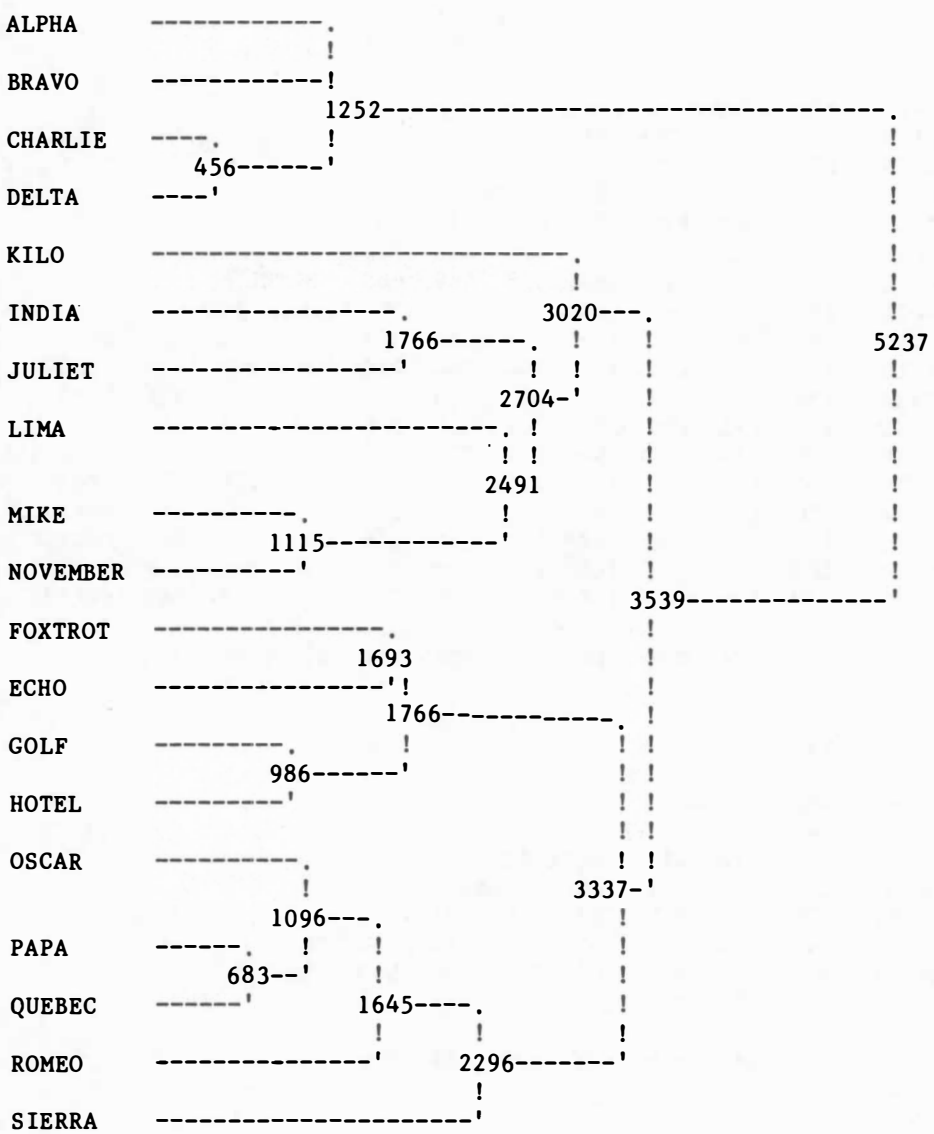
BRAVO	635	BRAVO									
CHARLIE	675	615	CHARLIE								
DELTA	635	590	825	DELTA							
ECHO	230	185	200	190	ECHO						
FOXTROT	200	160	190	175	490	FOXTROT					
GOLF	185	155	170	165	475	495	GOLF				
HOTEL	180	170	170	160	505	485	660	HOTEL			
INDIA	195	185	210	200	335	310	295	300	INDIA		
JULIET	190	160	185	175	355	350	335	350	475	JULIET	
KILO	110	125	125	115	240	250	250	250	315	415	KILO
LIMA	155	150	165	155	265	250	285	270	320	335	LIMA
MIKE	200	195	200	200	340	350	345	345	395	410	MIKE
NOVEMBER	190	185	205	190	275	295	305	295	330	385	NOVEMBER
OSCAR	150	160	160	145	275	275	285	285	270	335	OSCAR
PAPA	180	165	180	160	285	295	275	270	280	315	PAPA
QUEBEC	180	160	180	160	270	260	245	245	285	340	QUEBEC
ROMEO	160	135	155	140	265	270	260	250	260	315	ROMEO
SIERRA	170	170	175	140	320	300	295	290	275	360	SIERRA
	ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL	

KILO

LIMA	280	LIMA						
MIKE	340	375	MIKE					
NOVEMBER	300	350	625	NOVEMBER				
OSCAR	275	275	335	275	OSCAR			
PAPA	270	270	315	275	660	PAPA		
QUEBEC	275	280	310	285	630	750	QUEBEC	
ROMEO	260	245	310	250	525	500	500	ROMEO
SIERRA	275	225	325	305	435	390	395	380
	KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

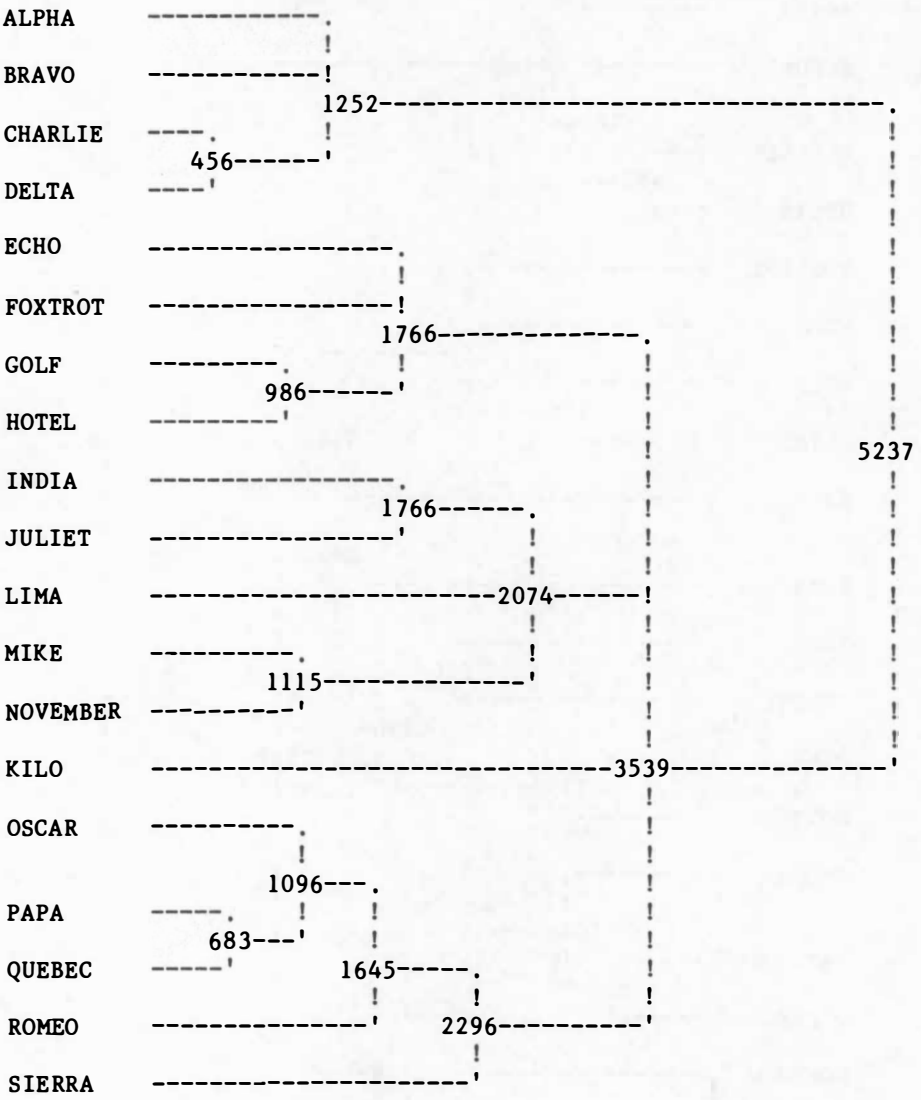
[810]

Eighth experiment: cognate percentages



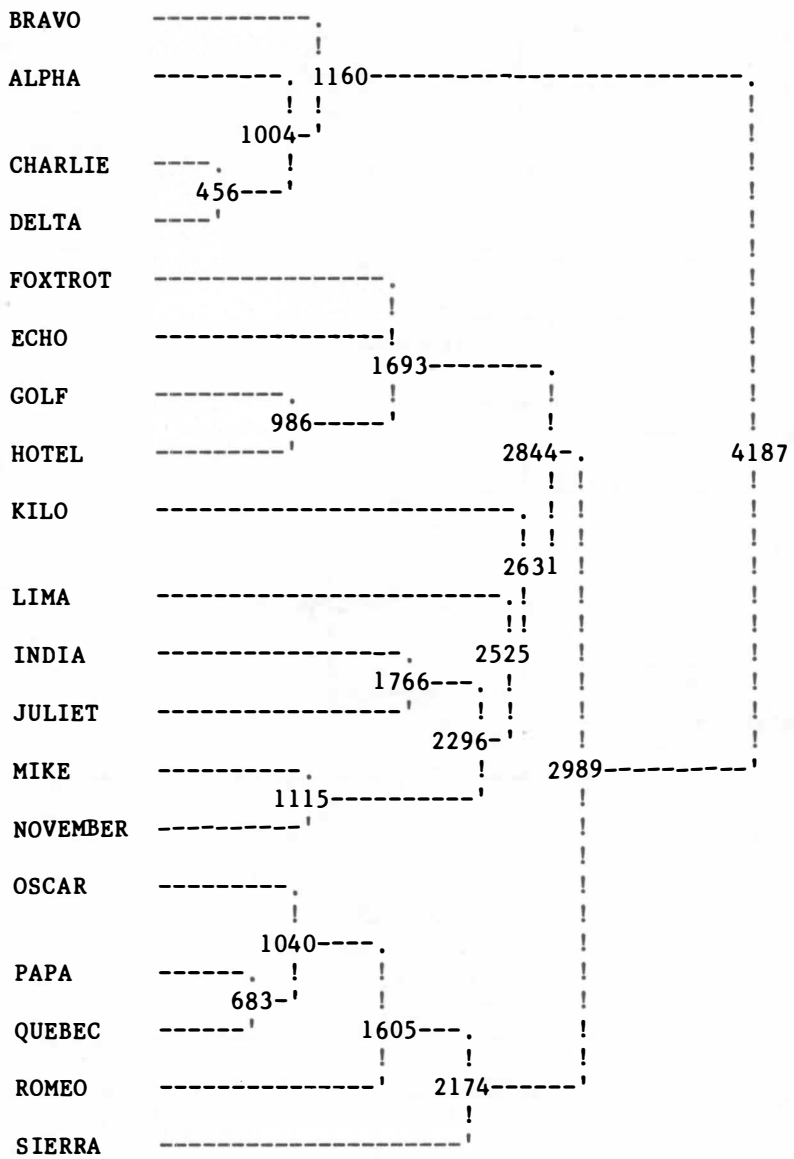
[821]

Eighth experiment: traditional lexicostatistical techniques, minimum-percentage method, no tolerance



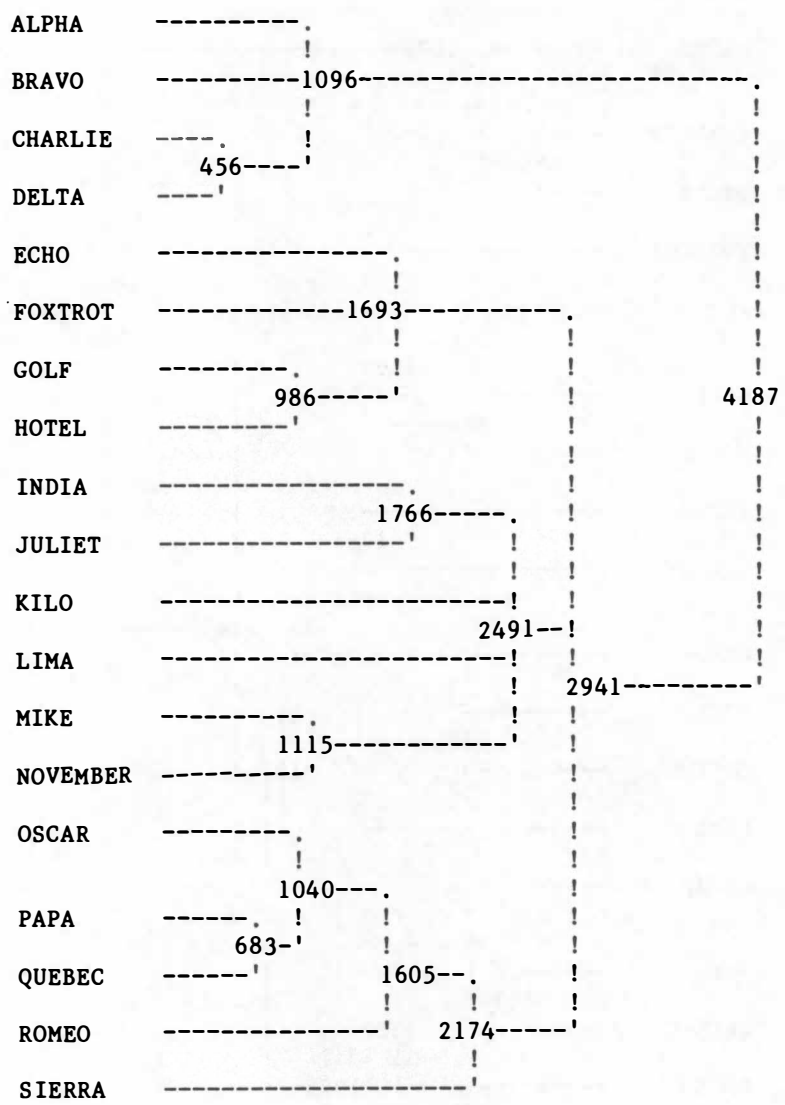
[822]

Eighth experiment: traditional lexicostatistical techniques, minimum-percentage method, 0.5 confidence level



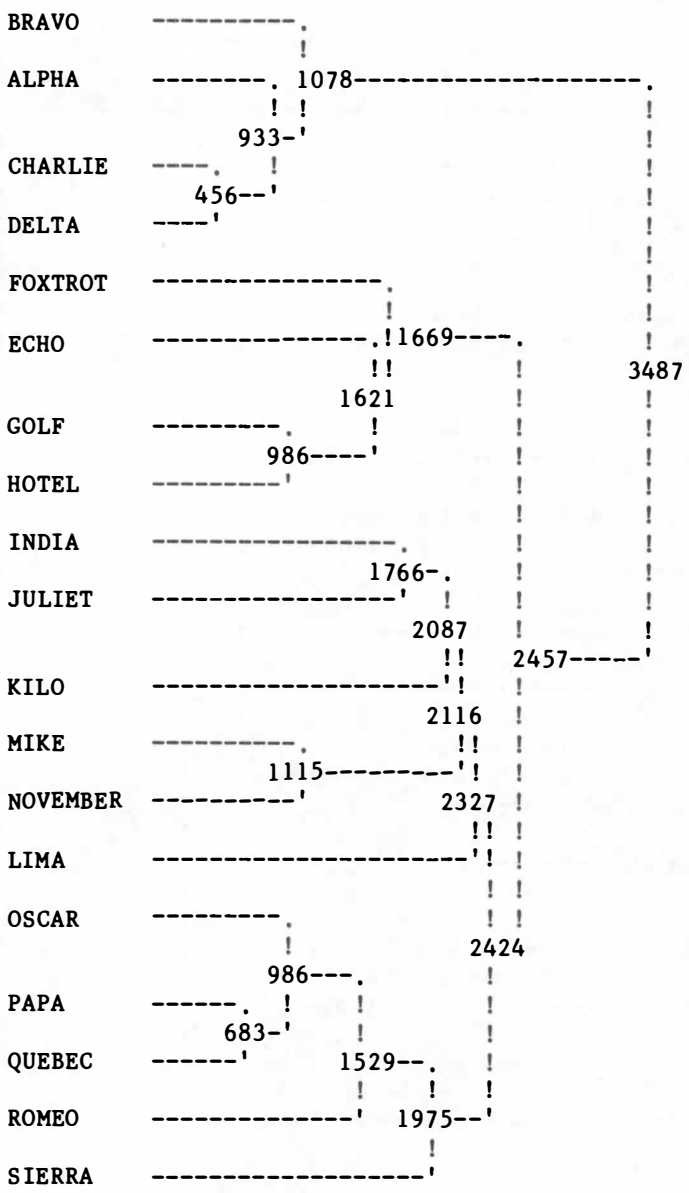
[823]

Eighth experiment: traditional lexicostatistical techniques, mean-percentage method, no tolerance



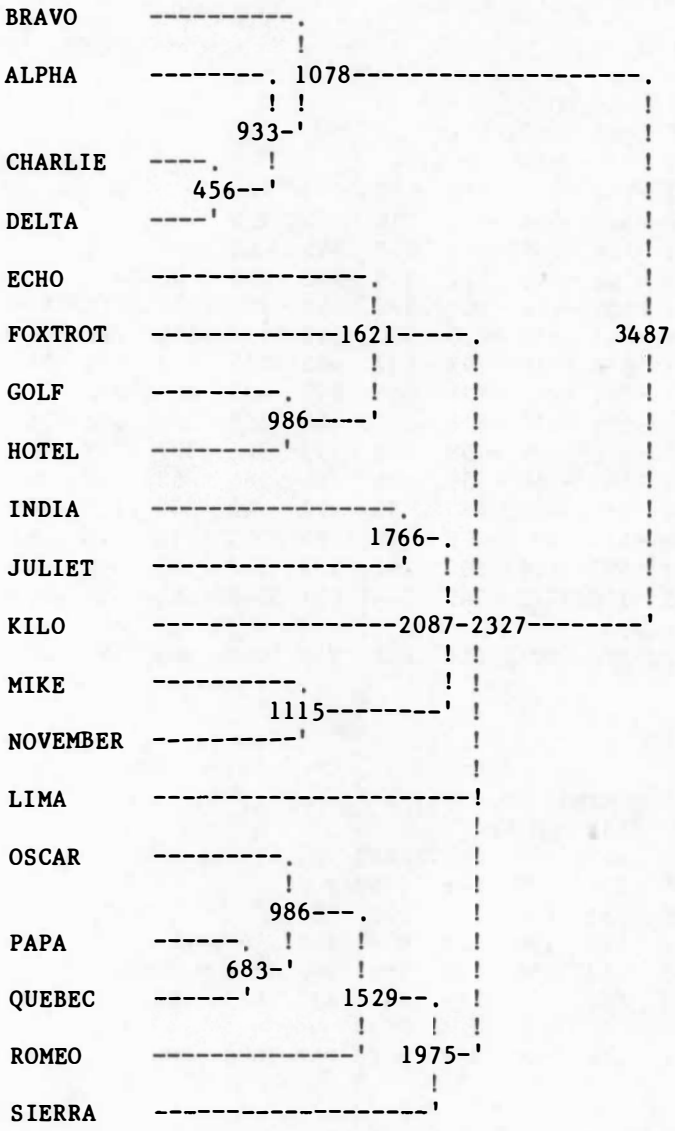
[824]

Eighth experiment: traditional lexicostatistical techniques, mean-percentage method, 0.5 confidence level



[825]

Eighth experiment: traditional lexicostatistical techniques, maximum-percentage method, no tolerance



[826]

Eighth experiment: traditional lexicostatistical techniques, maximum-percentage method, 0.5 confidence level

ALPHA

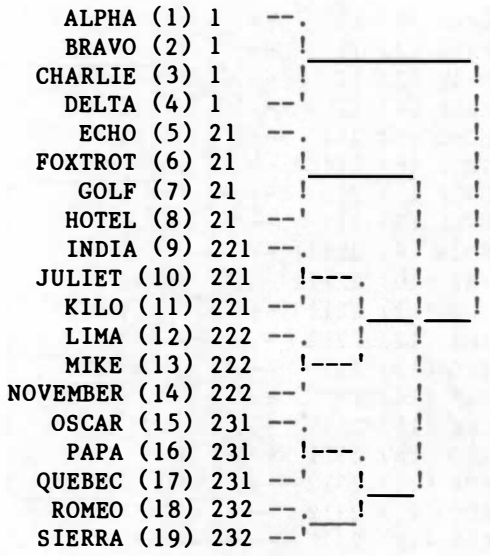
BRAVO	995	BRAVO									
CHARLIE	976	980	CHARLIE								
DELTA	980	978	999	DELTA							
ECHO	-513	-502	-512	-489	ECHO						
FOXTROT	-537	-540	-564	-531	986	FOXTROT					
GOLF	-491	-498	-509	-493	957	945	GOLF				
HOTEL	-464	-504	-496	-471	939	962	994	HOTEL			
INDIA	-631	-639	-644	-609	475	553	487	486	INDIA		
JULIET	-864	-843	-839	-820	500	542	497	461	919	JULIET	
KILO	-829	-832	-819	-797	467	492	415	412	918	947	KILO
LIMA	-772	-764	-763	-729	466	559	436	467	873	893	LIMA
MIKE	-646	-636	-624	-616	361	414	392	374	698	747	MIKE
NOVEMBER	-492	-477	-499	-459	378	392	326	337	775	662	NOVEMBER
OSCAR	-559	-569	-560	-559	116	161	84	68	246	368	OSCAR
PAPA	-504	-488	-493	-485	50	70	43	40	158	336	PAPA
QUEBEC	-500	-475	-485	-478	19	83	33	18	170	293	QUEBEC
ROMEO	-621	-593	-604	-600	161	197	138	143	310	446	ROMEO
SIERRA	-765	-770	-778	-745	344	431	348	356	534	603	SIERRA
	ALP	BRA	CHA	DEL	ECH	FOX	GOL	HOT	IND	JUL	

KILO

LIMA	917	LIMA						
MIKE	753	885	MIKE					
NOVEMBER	741	842	959	NOVEMBER				
OSCAR	495	378	197	206	OSCAR			
PAPA	420	331	168	134	970	PAPA		
QUEBEC	436	319	194	129	976	991	QUEBEC	
ROMEO	560	463	258	270	989	960	955	ROMEO
SIERRA	733	685	465	379	861	835	819	922
	KIL	LIM	MIK	NOV	OSC	PAP	QUE	ROM

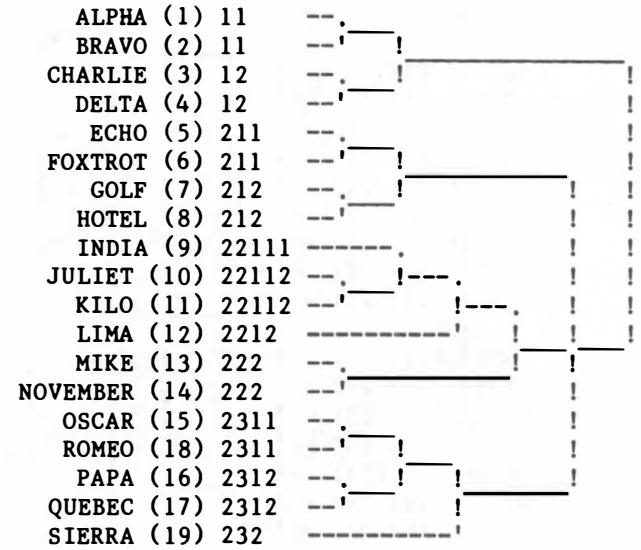
[830]

Eighth experiment: correlation coefficients



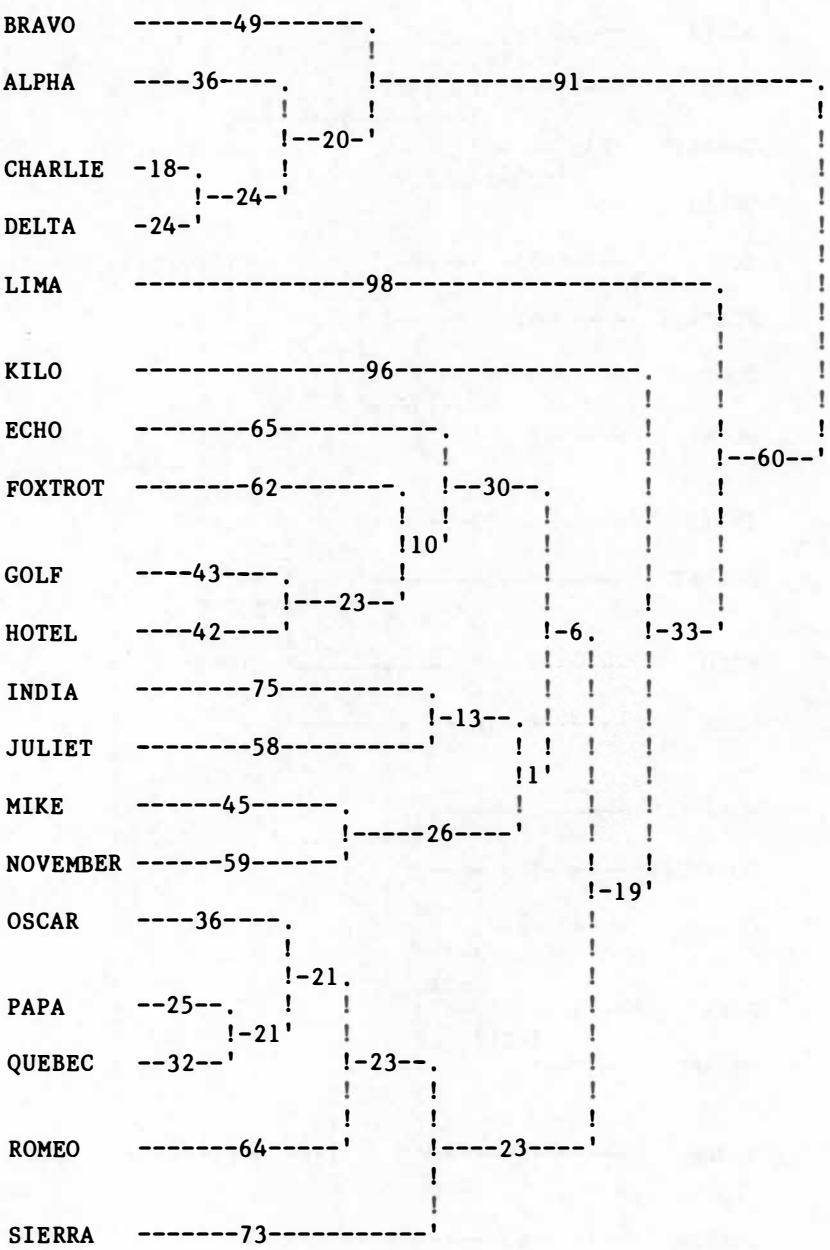
[847]

Eighth experiment: n-way splitting algorithm, with recomputations



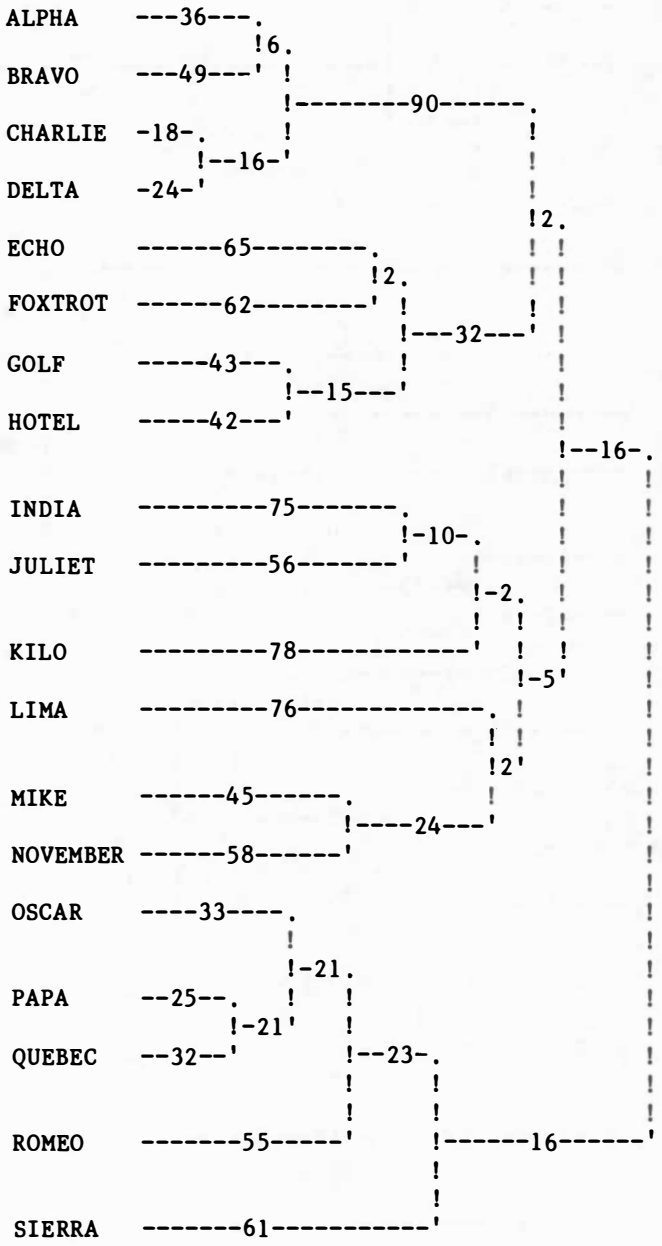
[848]

Eighth experiment: n-way splitting algorithm, no recomputations



[850]

Eighth experiment: reduced mutation algorithm



[860]

Eighth experiment: wild-card algorithm

APPENDIX 2

SOFTWARE

All programs given in this appendix are adapted from a subset of the author's personal software. They were written for DEC-KL10 in SIMULA and should run without modifications on any DEC10 or DEC20 machine, and with minimal or no modifications on any machine for which the Swedish National Defense Research Institute and the Norwegian Computing Center have made SIMULA compilers available. Although they were extensively tested and are believed to be free of bugs, the author declines all responsibilities regarding their use and no correspondance will be entered into on their subject.

PREPARING THE DATA FOR PROCESSING

COGNATE PERCENTAGES

The method to be followed in preparing a table of percentages of shared cognates for processing is best shown by an example.

Consider the table of percentages of shared cognates for 17 New Hebrides languages (Tryon 1973:308):

	1	2	3	4	5	6	. . .
1. Toga (Torres)	-	64	64	57	29	51	. . .
2. Mosina (Banks)	64	-	58	51	28	45	. . .
3. Peterara (Maewo)	64	58	-	65	34	55	. . .
4. Nduindui (Aoba)	57	51	65	-	65	52	. . .
5. Sakao (Santo)	29	28	34	32	-	40	. . .
6. Malo (Santo)	51	45	55	52	40	-	. . .
. . . . .	. . . . .						

The first line of the data file should contain the number 17 (the number of languages or wordlists) anywhere in the first six columns, the next 17 lines should contains the names (or some other kind of identification) of the languages in the same order as they appear in the table, and should be followed by the percentages under the diagonal, multiplied by 10, 10 to a line, each occupying 4 columns, e.g.:

```
17
Toga
Mosina
Peterara
. . . .
. . . .
Aneityum
 640 640 580 570 510 650 290 280 340 320
 510 450 550 520 400 390 390 450 410 430
  . . . . . .
  . . . . . .
 320 290 260 310 330 340
```

Language names more than 10 characters long (spaces included) will be shortened to their first 10 characters. Note that, being at most 3-digits long, each percentage is separated from the next on the line by at least one space. It is good practice to line them up as shown above, each one right-justified in its 4-column wide field. Make sure that no percentages are missing as the programs will abort if they find too few.

WORDLISTS TO BE PROCESSED BY THE WILD-CARD ALGORITHM

The first line of the file should contain the number of wordlists anywhere in columns 1 to 6 and the number of items in the list anywhere in columns 7 to 12. Next to follow are the corresponding language names, one per line. Finally, the wordlists themselves, in the same order as the language names were given, each new wordlist starting a new line and consisting of a list of forms, coded into numbers from -1 to 61, twenty numbers to a line, each number occupying 3 columns (each number is thus separated from the next on the same line by at leat one space).

The number coding is quite straightforward: a number from 0 up to 61 having been assigned to each of the different protoforms reconstructed for a given item, the gloss shown by the list for that item is replaced by the number identifying the protoform from which it is thought to have derived, missing items and loanwords being represented by -1. Any item for which more than 62 different protoforms can be reconstructed should be discarded from the lists.

Thus for example, taking just five words from Lehali, Tolomako, Sakao, and Shark-Bay we have, reduced to protoforms, and with an asterisk showing loanwords (Guy 1978):

	fish	dog	rat	sea	bow
Lehali	iga	*dog	gazuwe	nawo	wusu
Tolomako	mazi	kwesi	garivi	tasi	liua
Sakao	mazi	*kwesi	aw(aeo)	tasi	paga
Shark-Bay	mazi	*kwesi	aw(aeo)	tasi	paga

which would be coded as:

```
4      5
Lehali
Tolomako
Sakao
Shark-bay
 00 -1 00 00 00
 01 00 01 01 01
 01 -1 02 01 02
 01 -1 02 01 02
```

There is strictly no need to input 0, 1, 2 etc. as 00, 01, 02, etc., but lining the numbers up is made easier this way.

RUNNING THE PROGRAMS

GENERAL PRINCIPLES

All the programs start by asking the name of the file where the data is to be found, and go on to ask under what name the results of the computations should be filed. Some programs ask more questions before they start processing the data.

PROGRAM LINEAR

LINEAR computes linear-correlation coefficients from a table of percentages prepared as described above.

PROGRAM LAYOUT

The tables of percentages and correlation coefficients in appendix 1 were produced by LAYOUT from the original computer-produced tables. Use LAYOUT to translate into a format fit for human consumption the cognate-percentage files prepared as described above and the output from LINEAR. LAYOUT asks one extra question: the maximum number of lines that can fit on a page of output.

PROGRAM LXSTAT

LXSTAT applies the traditional lexicostatistical technique, mean-percentage method, on the cognate percentages prepared as described earlier. It asks several extra questions; first for an estimate of the retention rate (to be given as a real number from 0 to 1, not as a percentage; thus 81% is to be given as 0.81), then for a tolerance (again to be given as a real number from 0 to 1; thus a tolerance of 4 percentage points is to be given as 0.04). If, when asked for the tolerance, you give a negative value, e.g. -1, LXSTAT understands that you want tolerances computed to a certain confidence level and goes on to ask for the confidence level required and the number of items in the wordlists (i.e. the number of items compared for each language pair).

PROGRAM SPLIT

Given a table of correlation coefficients (computed by LINEAR), SPLIT uses the n-way splitting algorithm to split the corresponding language family into subgroups. Nothing prevents you from giving SPLIT the original file of cognate percentages to work on, but of course, the results will be quite different.

PROGRAM TREE

TREE produces information for drawing a minimum-spanning tree using the single-linkage method from a file of cognate percentages or of correlation coefficients.

PROGRAM WILDC

WILDC applies the wild-card algorithm to wordlists coded as described page 182.

A COMPLETE EXAMPLE

In the detailed example which follows it is assumed that the computer used is a DEC-10, that the program listings have been filed under the names LAYOUT.SIM, LXSTAT.SIM, LINEAR.SIM, etc., and have been already compiled and saved using the monitor commands COMPILE, LOAD, and SAVE or OSAVE.

THE PERCENTAGE FILE

The cognate percentages are taken from Tryon 1973:308. The whole data file, prepared using EDIT, TECO, VIDED, or whichever editor is most convenient, is shown on page 185. Call this file NH17.PC (of course, almost any other name would do).

RUNNING LAYOUT

When, to have NH17.PC translated into a readable tabular form, you run LAYOUT, the following exchange takes place between you and the DEC-10 (your contribution to the dialogue is underlined).

```
.RUN LAYOUT

Table is in file: NH17.PC
Edited table goes to file: NH17.TAB
Lines per page: 50

3 garbage collection(s) in 0 ms

End of SIMULA program execution.
CPU time: 0.20  Elapsed time: 0.70
```

The contents of the file created by LAYOUT, NH17.TAB, are shown on page 187.

```
17
Toga
Mosina
Peterara
Nduindui
Sakao
Malo
Fortsenal
Raga
Sa
Dakaka
Aulua
Big Nambas
Lewo
Nguna
Sie
Lenakel
Aneityum
 640 640 580 570 510 650 290 280 340 320
 510 450 550 520 400 390 390 450 410 430
 500 520 480 570 600 310 480 450 430 420
 480 510 280 450 390 470 400 420 430 430
 270 370 300 380 490 430 410 440 440 260
 380 350 440 450 470 360 310 390 380 250
 330 290 380 360 340 450 400 360 410 370
 200 360 300 400 400 320 390 350 470 370
 390 420 250 420 390 430 420 410 480 400
 360 310 330 310 310 250 280 270 300 310
 290 280 270 280 320 290 270 250 270 150
 220 230 260 240 310 310 250 240 290 280
 350 320 330 360 210 300 290 340 300 320
 320 290 260 310 330 340
```

Percentages of shared cognates between 17 New Hebrides languages prepared for computer processing

```
RUNNING LXSTAT

   The following exchange takes place:

     .RUN LXSTAT

     Percentages are in file: NH17.PC
     Results go to file: NH17.LEX
     Retention rate: 0.81
     Tolerance: 0.04

     2 garbage collection(s) in 0 ms

     End of SIMULA program execution.
     CPU time: 0.37  Elapsed time: 0.90

   Here are the contents of file NH17.LEX:

   Retention rate: 0.8100. Tolerance: 0.0400
   / Peterara
   / Nduindui
   / Toga
     0.62000 1134 years.
   / Mosina
   / Peterara - Nduindui - Toga
   / Raga
     0.55714 1388 years.
   / Mosina - Peterara - Nduindui - Toga - Raga
   / Malo
   / Fortsenal
     0.46364 1824 years.
   / Sa
   / Dakaka
   / Mosina - Peterara - Nduindui - Toga - Raga - Malo - Fortsenal
     0.42467 2032 years.
   / Aulua
   / Nguna
   / Big Nambas
     0.44333 1930 years.
   / Sa - Dakaka - Mosina - Peterara - Nduindui - Toga - Raga - Malo -
   Fortsenal
   / Aulua - Nguna - Big Nambas
   / Lewo
     0.38692 2253 years.
   / Lenakel
   / Aneityum
   / Sa - Dakaka - Mosina - Peterara - Nduindui - Toga - Raga - Malo -
   Fortsenal - Aulua - Nguna - Big Nambas - Lewo
   / Sie
     0.29357 2908 years.
   / Sakao
   / Lenakel - Aneityum - Sa - Dakaka - Mosina - Peterara - Nduindui -
```

Toga

Mosina	640	Mosina									
Peterara	640	580	Peterara								
Nduindui	570	510	650	Nduindui							
Sakao	290	280	340	320	Sakao						
Malo	510	450	550	520	400	Malo					
Fortsenal	390	390	450	410	430	500	Fortsenal				
Raga	520	480	570	600	310	480	450	Raga			
Sa	430	420	480	510	280	450	390	470	Sa		
Dakaka	400	420	430	430	270	370	300	380	490	Dakaka	
Aulua	430	410	440	440	260	380	350	440	450	470	Aulua
Big Nambas	360	310	390	380	250	330	290	380	360	340	Big Nambas
Lewo	400	360	410	370	200	360	300	400	400	320	Lewo
Nguna	470	370	390	420	250	420	390	430	420	410	Nguna
Sie	310	330	310	310	250	280	270	300	310	290	Sie
Lenakel	290	270	250	270	150	220	230	260	240	310	Lenakel
Aneityum	350	320	330	360	210	300	290	340	300	320	Aneityum
	Tog	Mos	Pet	Ndu	Sak	Mal	For	Rag	Sa	Dak	

Aulua

Big Nambas	450	Big Nambas				
Lewo	390	350	Lewo			
Nguna	480	400	360	Nguna		
Sie	280	270	280	320	Sie	
Lenakel	310	250	240	290	280	Lenakel
Aneityum	320	290	260	310	330	340	Aneityum
	Aul	Big	Lew	Ngu	Sie	Len

Output from program LAYOUT

Toga - Raga - Malo - Fortsenal - Aulua - Nguna - Big Nambas - Lewo - Sie
0.28063 3015 years.

From this output a genealogical tree can easily be drawn (page 189).

RUNNING LINEAR

The following exchange takes place:

```
.RUN LINEAR

Cognate percentages are in file: NH17.PC
Linear-correlation coefficients go to file: NH17.LIN

2 garbage collection(s) in 0 ms

End of SIMULA program execution.
CPU time: 0.27  Elapsed time: 0.72
```

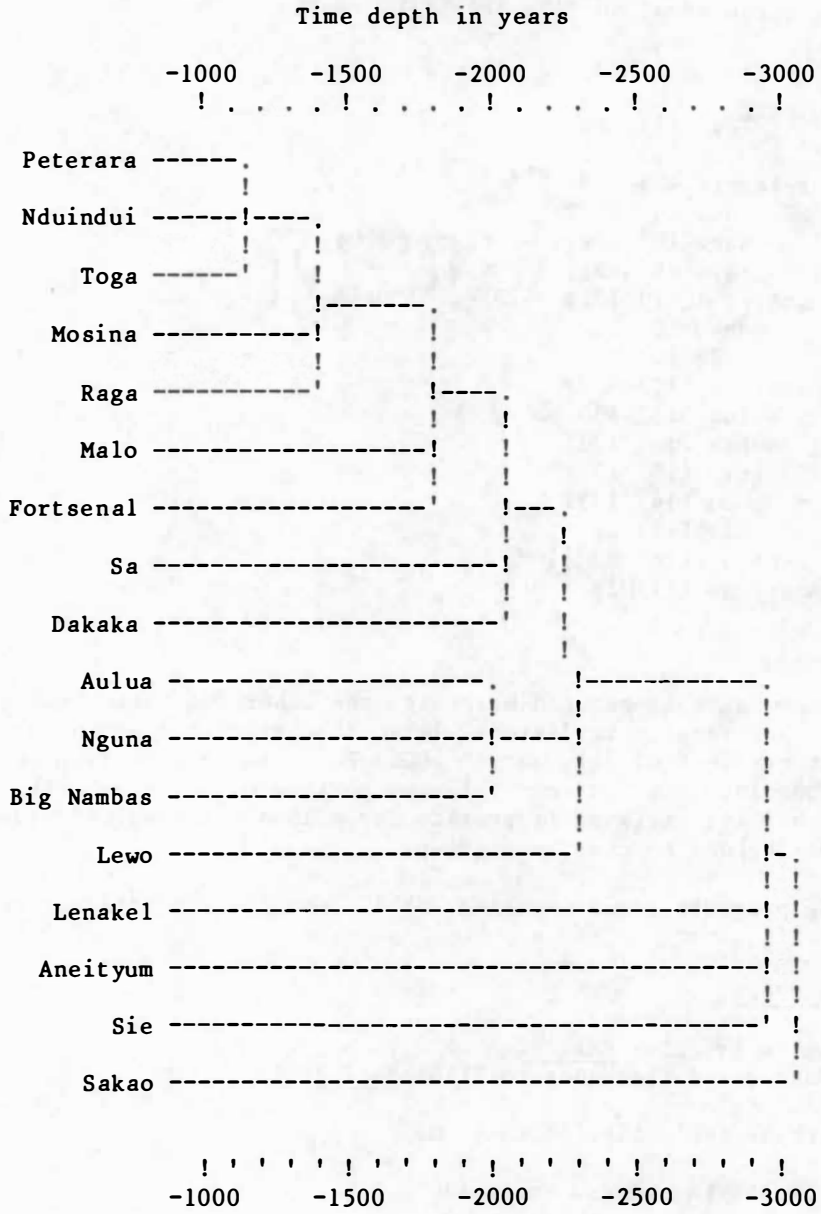
The resulting file, NH17.LIN, can be translated into a table fit for human consumption by running LAYOUT. Thus:

```
.RUN LAYOUT

Table is in file: NH17.LIN
Edited table goes to file: NH17.COR
Lines per page: 50

3 garbage collection(s) in 0 ms

End of SIMULA program execution.
CPU time: 0.20  Elapsed time: 0.72
```

RUNNING SPLIT

The following exchange takes place:

```
.RUN SPLIT

Table is in file: NH17.LIN
Reconstructed tree goes to file: NH17.SPL

3 garbage collection(s) in 41 ms

End of SIMULA program execution.
CPU time: 0.38  Elapsed time: 0.80
```



Seventeen New Hebrides languages grouped by traditional lexicostatistical techniques, mean-percentage method (tree drawn from output from program LXSTAT)

The results, contained in file NH17.SPL, read:

```
          Toga (1) 111
        Mosina (2) 111
     Peterara (3) 1121
     Nduindui (4) 1122
         Raga (8) 1122
         Malo (6) 121
    Fortsenal (7) 121
        Sakao (5) 122
           Sa (9) 131
      Dakaka (10) 132
       Aulua (11) 1331
  Big Nambas (12) 1331
        Lewo (13) 1332
       Nguna (14) 1332
         Sie (15) 21
     Lenakel (16) 22
   Aneityum (17) 22
```

Note that Sakao is grouped here with the other two Santo languages, Malo and Fortsenal. Traditional lexicostatistics had shown Sakao to have split earliest of all, about 1000 BC. In fact, despite its unusual phonology, its extremely low proportion of cognates with other languages and its striking propensity for holophrastic verbal clauses, Sakao does belong to the Santo group.

Nothing prevents you from using SPLIT on the cognate-percentage table itself, thus:

```
.RUN SPLIT

Table is in file: NH17.PC
Reconstructed tree goes to file: NH17.XXX

3 garbage collection(s) in 47 ms

End of SIMULA program execution.
```

The reconstruction (shown on the next page) obtained by running SPLIT with the table of cognate percentages as input is, not surprisingly, rather similar to the reconstructions which can be obtained through traditional lexicostatistical techniques.

```
          Toga (1) 111
        Mosina (2) 111
      Peterara (3) 1121
      Nduindui (4) 1121
          Raga (8) 1122
          Malo (6) 113
    Fortsenal (7) 12
       Aulua (11) 13
       Nguna (14) 13
           Sa (9) 14
      Dakaka (10) 14
        Sakao (5) 2
  Big Nambas (12) 3
        Lewo (13) 4
         Sie (15) 5
    Lenakel (16) 6
   Aneityum (17) 7
```

RUNNING TREE

The following exchange takes place:

```
.RUN TREE

Table is in file: NH17.PC
Minimum-spanning tree goes to file: NH17.TRE

2 garbage collection(s) in 0 ms

End of SIMULA program execution.
CPU time: 0.16  Elapsed time: 1.76
```

The resulting file contains information for the construction of a minimum-spanning tree by the single-linkage method:

```
0.650 Peterara - Nduindui
0.640 Peterara - Toga
0.640 Toga - Mosina
0.600 Nduindui - Raga
0.550 Peterara - Malo
0.510 Nduindui - Sa
0.500 Malo - Fortsenal
0.490 Sa - Dakaka
0.470 Toga - Nguna
0.480 Nguna - Aulua
0.450 Aulua - Big Nambas
0.430 Fortsenal - Sakau
0.410 Peterara - Lewo
0.360 Nduindui - Aneityum
0.340 Aneityum - Lenakel
0.330 Mosina - Sie
```

The corresponding tree is shown on page 193. This tree would be the most likely map of the migrations of the speakers of these languages if, when communities split, those who migrate did start replacing vocabulary until they settle in their new location and those who stay never split again and never replace any vocabulary. Not a very plausible model.

Nothing prevents you from using TREE on the table of linear-correlation coefficients produced by LINEAR. The resulting tree would be, not a genealogical tree, but a graph showing the genetic closeness of these languages.

RUNNING WILDC

The four 5-item wordlists (Lehali, Tolomako, Sakao, Shark-Bay) have been filed under the name NH17.LST. The following exchange takes place:

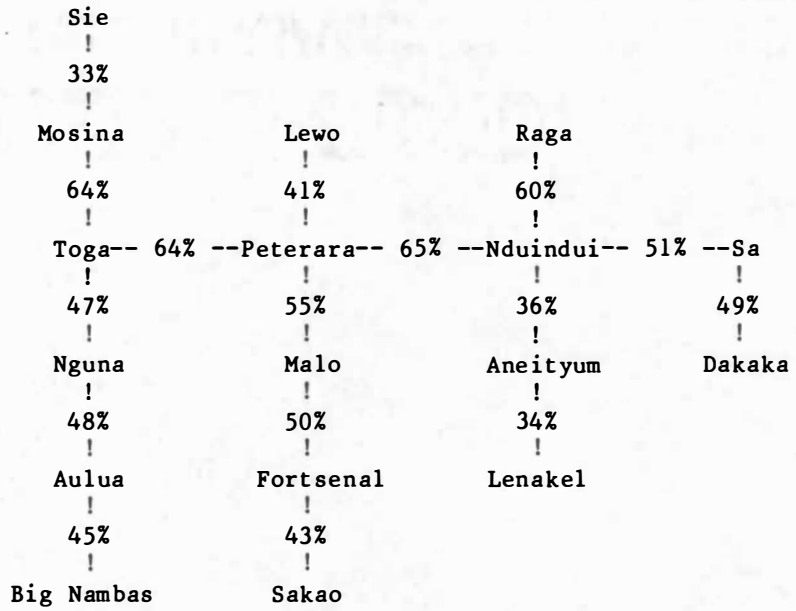
```
.RUN WILDC

Wordlists (reduced to cognate groups) are in file: NH17.LST
Reconstruction goes to file: NH17.WIL

2 garbage collection(s) in 0 ms

End of SIMULA program execution.
CPU time: 0.14  Elapsed time: 0.80
```

The results, in NH17.WIL, read:

```
Group#1 = Lehali, Tolomako, Sakao, Shark-bay
1.212
Lehali replaced 5 items:  1  2  3  4  5
*****
Tolomako replaced 3 items:  2  3  5
1**1*
Sakao replaced 1 item:  2
1*212
Shark-bay replaced 1 item:  2
1*212
```

Interpreting the results

The wordlist of the protolanguage is reconstructed as consisting of protoform 1, a form which cannot be reconstructed for lack of evidence (symbolized by a dot), protoform 2, protoform 1, and protoform 2. So,



Minimum-spanning tree, single-linkage method, from percentages of shared cognates between 17 New Hebrides languages, drawn from output of program TREE

according to the reconstruction, the protolanguage had:

fish: mazi
dog: (unknown)
rat: aw(aeo)
sea: tasi
bow: paga

If there had been for some item more than ten protoforms, protoforms 10 to 35 would have be represented in the output from WILDC by capital letters (A to Z) and protoforms 36 to 61 by small letters (a to z).

PROGRAM LISTINGS

LINEAR

```
BEGIN
    REF(Infile) raw;
    REF(Outfile) cooked;
    TEXT rawimage,cookedimage;
    INTEGER z,i;
    rawimage:-Blanks(80);
    cookedimage:-Blanks(40);
    Outtext("Cognate percentages are in file: ");
    Breakoutimage; Inimage;
    raw:-NEW Infile(Sysin.Image.Strip);
    Outtext("Linear-correlation coefficients go to file: ");
    Breakoutimage; Inimage;
    cooked:-NEW Outfile(Sysin.Image.Strip);
    cooked.Open(cookedimage);
    INSPECT raw DO
    BEGIN
        Open(rawimage);
        Inimage;
        z:=Inint;
        cooked.Outint(z,6);
        cooked.Outimage;
        FOR i:=1 STEP 1 UNTIL z DO
        BEGIN
            Inimage; cooked.Outtext(rawimage.Sub(1,40));
            cooked.Outimage
        END
    END;

    BEGIN
        INTEGER i,j,k,l,n;
        TEXT ARRAY rawsub[1:10];
        REAL ARRAY t,tr[1:z,1:z];

        FOR i:=1 STEP 1 UNTIL 10 DO
        rawsub[i]:-rawimage.Sub(i*4-3,4);

        n:=z; k:=0; raw.Inimage;
        FOR i:=2 STEP 1 UNTIL n DO
        BEGIN
            l:=i-1;
            FOR j:=1 STEP 1 UNTIL l DO
```

```
            BEGIN
                k:=k+1;
                IF k>10 THEN
                BEGIN
                    k:=1; raw.Inimage
                END;
                t[i,j]:=t[j,i]:=rawsub[k].Getint/1000
            END
        END;
        raw.Close;
        FOR i:=1 STEP 1 UNTIL n DO
        FOR j:=i+1 STEP 1 UNTIL n DO
        BEGIN
            REAL sxy,sx,sy,sx2,sy2,x,y,sxyn,sx2n,sy2n,r;
            INTEGER k,fin,nn;
            nn:=n-2; fin:=n;
            FOR k:=1 STEP 1 UNTIL fin DO
            IF k\=i AND k\=j THEN
            BEGIN
                y:=t[i,k]; x:=t[j,k];
                sx:=sx+x; sy:=sy+y;
                sxy:=sxy+x*y;
                sx2:=sx2+x*x;
                sy2:=sy2+y*y
            END;
            sxyn:=nn*sxy-sx*sy;
            sx2n:=nn*sx2-sx**2;
            sy2n:=nn*sy2-sy**2;
            r:= IF Abs(sxyn/nn**2)<1.0&-8 THEN 0.0 ELSE
            sxyn/Sqrt(sx2n*sy2n);
            tr[i,j]:=r; tr[j,i]:=r;
        END;

        FOR i:=2 STEP 1 UNTIL n DO
        BEGIN
            l:=i-1;
            FOR j:=1 STEP 1 UNTIL l DO
            cooked.Outint(Entier(tr[i,j]*1000+0.5),4)
        END;
        cooked.Close
    END
END
```

LAYOUT

```
BEGIN
    INTEGER z;
    REF(Infile) raw;
    TEXT rawimage;
    rawimage:-Blanks(80);
    Outtext("Raw data is in file: ");
    Breakoutimage; Inimage;
    raw:-NEW Infile(Sysin.Image.Strip);
    raw.Open(rawimage);
    raw.Inimage;
    z:=rawimage.Sub(1,6).Getint;

    BEGIN
        INTEGER i,j,k,l,n,top,left,right,rightmost,
        pagesize;
        TEXT margin,cookedline,headline;
        TEXT ARRAY headsub,cookedsub,rawsub[1:10],
        cookedright[1:11],langname,lang3,leftlang[1:z];
        INTEGER ARRAY t[1:z,1:z];
        REF(Printfile) cooked;

        n:=z;
        Outtext("Edited data goes to file: ");
        Breakoutimage; Inimage;
        cooked:-NEW Printfile(Sysin.Image.Strip);
        Outtext("Lines per page: "); Breakoutimage; Inimage;
        pagesize:=Inint;
        cookedline:-Blanks(78);
        headline:-Blanks(78);
        FOR i:=1 STEP 1 UNTIL 10 DO
        BEGIN
            cookedsub[i]:-cookedline.Sub(i*5+10,4);
            rawsub[i]:-rawimage.Sub(i*4-3,4);
            headsub[i]:-headline.Sub(i*5+11,3);
            cookedright[i]:-cookedline.Sub(i*5+12,10)
        END;
        cookedright[11]:-cookedline.Sub(66,10);
        margin:-cookedline.Sub(1,12);
        FOR i:=1 STEP 1 UNTIL n DO
        BEGIN
            INTEGER lgth; TEXT lang;
            raw.Inimage;
            lgth:=rawimage.Strip.Length;
            IF lgth>10 THEN lgth:=10;
            lang:-rawimage.Sub(1,lgth);
            langname[i]:-Copy(lang);
```

```
    leftlang[i]:-Blanks(12);
    leftlang[i].Sub(13-lgth,lgth):=lang.Sub(1,lgth);
    lang3[i]:-Copy(IF lgth<4 THEN lang ELSE lang.Sub(1,3));
END;

k:=0; raw.Inimage;
FOR i:=2 STEP 1 UNTIL n DO
BEGIN
    l:=i-1;
    FOR j:=1 STEP 1 UNTIL l DO
    BEGIN
        k:=k+1;
        IF k>10 THEN
        BEGIN
            k:=1; raw.Inimage
        END;
        t[i,j]:=rawsub[k].Getint
    END
END;
raw.Close;
cooked.Open(cookedline);
cooked.Linesperpage(pagesize);

top:=2;
INSPECT cooked DO
BEGIN
    `nextcol:
    Eject(1);
    cookedline:=NOTEXT;
    cookedline.Sub(16,12):=langname[top-1];
    Outimage; Outimage;
    left:=top-1; rightmost:=left+9;
    IF rightmost>=n THEN rightmost:=n-1;
    k:=0;
    headline:=NOTEXT;
    FOR i:=left STEP 1 UNTIL rightmost DO
    BEGIN
        k:=k+1;  headsub[k]:=lang3[i]
    END;
    FOR i:=top STEP 1 UNTIL n DO
    BEGIN
        right:=IF i>rightmost THEN rightmost ELSE i-1;
        k:=0;
        FOR j:=left STEP 1 UNTIL right DO
        BEGIN
            k:=k+1;
            cookedsub[k].Putint(t[i,j]);
        END;
        cookedright[k+1]:=langname[i];
        margin:=leftlang[i];
        Outimage;
    END;
```

```
            IF rightmost<n-1 THEN
            BEGIN
                IF Line<pagesize THEN
                BEGIN
                    Outimage;
                    Outtext(headline.Sub(1,13+5*(right-left+1)));
                    Outimage
                END;
                top:=rightmost+2;
                GOTO `nextcol
            END;
            IF Line<pagesize THEN
            BEGIN
                Outimage;
                Outtext(headline.Sub(1,13+5*(right-left+1)));
                Outimage
            END
        END;

        cooked.Close
    END
END
```

LXSTAT

```
BEGIN
    REF(Infile) raw;
    REF(Outfile) cooked;
    TEXT rawimage,cookedimage;
    INTEGER z,i,size;
    REAL r,e,zscore;
    rawimage:-Blanks(80);
    cookedimage:-Blanks(70);
    Outtext("Percentages are in file: ");
    Breakoutimage; Inimage;
    raw:-NEW Infile(Sysin.Image.Strip);
    Outtext("Results go to file: ");
    Breakoutimage; Inimage;
    cooked:-NEW Outfile(Sysin.Image.Strip);
    Outtext("Retention rate: ");
    Breakoutimage; Inimage;
    r:=Inreal;
    INSPECT cooked DO
    BEGIN
        Open(cookedimage);
        Outtext("Retention rate:");
        Outfix(r,4,7);
    END;
    Outtext("Tolerance: "); Breakoutimage; Inimage;
    e:=Inreal;
    IF e>=0.0 THEN
    BEGIN
        INSPECT cooked DO
        BEGIN
            Outtext(". Tolerance:");
            Outfix(e,4,7);
            Outimage
        END
    END
    ELSE
    BEGIN
        Outtext("Number of items: ");
        Breakoutimage; Inimage;
        size:=Inint;
        Outtext("Confidence level: ");
        Breakoutimage; Inimage;
        BEGIN
            REAL  q,t;
            q:=(1.0-Inreal)/2;
            t:=Sqrt(Ln(1/q**2));
```

```
        zscore:= t-(
        (2.515517+0.802853*t+0.010328*t**2)
        /(1.0+1.432788*t+0.189269*t**2+0.001308*t**3)
        );
        INSPECT cooked DO
        BEGIN
            Outtext(".  Wordlists: ");
            Outint(size,4);
            Outtext(" items.");
            Outimage;
            Outtext("Level of confidence:");
            Outfix(1.0-2*q,5,8);
            Outtext(" (");
            Outfix(zscore,5,7);
            Outtext(" standard deviations).");
            Outimage
        END
    END;
END;
INSPECT raw DO
BEGIN
    Open(rawimage);
    Inimage;
    z:=Inint;
END;

BEGIN
    INTEGER i,j,k,l,n,remain;
    TEXT ARRAY rawsub[1:10],langname[1:z];
    REAL ARRAY pc[1:z,1:z];
    BOOLEAN ARRAY assigned[1:z];
    INTEGER ARRAY next[1:z];
    n:=z;
    FOR i:=1 STEP 1 UNTIL 10 DO
    rawsub[i]:-rawimage.Sub(i*4-3,4);

    INSPECT raw DO
    BEGIN
        FOR i:=1 STEP 1 UNTIL n DO
        BEGIN
            Inimage;
            langname[i]:-Copy(Image.Strip)
        END;


        Inimage;
        FOR i:=2 STEP 1 UNTIL n DO
        BEGIN
            l:=i-1;
            FOR j:=1 STEP 1 UNTIL l DO
            BEGIN
                k:=k+1;
```

```
                IF k>10 THEN
                BEGIN
                    k:=1; Inimage
                END;
                pc[i,j]:=pc[j,i]:=rawsub[k].Getint/1000
            END
        END;
        Close
    END;
    remain:=n;
    WHILE remain>1 DO
    BEGIN
        REAL u,v,umax;
        INTEGER i,j,l,imax,jmax,cases;
        BOOLEAN ARRAY join[1:n];

        BOOLEAN  PROCEDURE same(p1,n1,p2,n2,z);
        INTEGER n1,n2; REAL p1,p2,z;
        BEGIN
            REAL p,sd;
            p:=(p1*n1+p2*n2)/(n1+n2);
            sd:=Sqrt(p*(1.0-p)*(1/n1+1/n2));
            same:=Abs(p1-p2)/sd<=z
        END;

        PROCEDURE append(i,j,cases0,score0);
        NAME cases0,score0;
        INTEGER i,j,cases0;
        REAL score0;
        BEGIN
            INTEGER k,nucases;
            REAL nuscore;
            nuscore:=score(i,j,nucases);
            score0:=(score0*cases0+nuscore*nucases)
            /(cases0+nucases);
            WHILE next[j]\=0 DO j:=next[j];
            next[j]:=i;
            assigned[i]:=TRUE;
            cases0:=cases0+nucases
        END;

        REAL PROCEDURE score(i,j,cases);
        NAME cases;
        INTEGER i,j,cases;
        IF NOT assigned[i] AND NOT assigned[j] THEN
        BEGIN
            REAL x,y; INTEGER k,n;
            WHILE i\=0 DO
            BEGIN
                k:=j;
                WHILE k\=0 DO
                BEGIN
```

```
            x:=pc[i,k];
            y:=y+x;
            n:=n+1;
            k:=next[k]
        END;
        i:=next[i]
    END;
    cases:=n;
    score:=y/n
END;

PROCEDURE report(f,i);
REF(Outfile) f;
INTEGER i;
INSPECT f DO
BEGIN
    Outtext("/ ");
    WHILE i\=0 DO
    BEGIN
        Outtext(langname[i]);
        i:=next[i];
        IF i\=0 THEN Outtext(" - ")
    END;
    IF Image.Strip=/=NOTEXT THEN Outimage
END;

FOR i:=2 STEP 1 UNTIL n DO
BEGIN
    l:=i-1;
    FOR j:=1 STEP 1 UNTIL l DO
    BEGIN
        u:=score(i,j,cases);
        IF u>umax THEN
        BEGIN  umax:=u; imax:=i; jmax:=j;
        END
    END
END;

FOR i:=1 STEP 1 UNTIL n DO
IF i\=imax AND i\=jmax AND NOT assigned[i] THEN
BEGIN
    BOOLEAN joinit;
    INTEGER ucases,vcases;
    u:=score(i,imax,ucases);
    v:=score(i,jmax,vcases);
    joinit:=
    IF size>0 THEN
    same(u,size,umax,size,zscore)
    OR
    same(v,size,umax,size,zscore)
    ELSE
    Abs(u-umax)<=e OR Abs(v-umax)<=e;
```

```
                IF joinit AND (e>0.0 OR zscore>0.0) THEN
                FOR j:=1 STEP 1 UNTIL n DO
                IF j\=i AND j\=imax AND j\=i AND NOT assigned[j] THEN
                BEGIN  REAL w; INTEGER wcases;
                    w:=score(i,j,wcases);
                    IF w>u AND w>v THEN
                    BEGIN
                        joinit:=FALSE; GOTO `exit
                    END;
                END;
                `exit:
                IF joinit THEN join[i]:=TRUE
            END;
            k:=0;
            report(cooked,jmax);
            report(cooked,imax);
            umax:=0.0;
            append(imax,jmax,k,umax);
            FOR i:=1 STEP 1 UNTIL n DO
            IF join[i] THEN
            BEGIN
                report(cooked,i);
                append(i,jmax,k,umax);
            END;
            INSPECT cooked DO
            BEGIN
                Outfix(umax,5,9);
                Outint(Entier(1000*Ln(umax)/(Ln(r)*2)+0.4999),5);
                Outtext(" years.");
                Outimage
            END;
            remain:=0;
            FOR i:=1 STEP 1 UNTIL n DO
            IF NOT assigned[i] THEN remain:=remain+1;
        END;
        cooked.Close
    END
END
```

SPLIT

```
BEGIN
    REF(Infile) raw;
    TEXT rawimage;
    INTEGER langnumber;
    Outtext("Table is in file: ");
    Breakoutimage; Inimage;
    raw:-NEW Infile(Sysin.Image.Strip);
    rawimage:-Blanks(80);
    INSPECT raw DO
    BEGIN
        Open(rawimage);
        Inimage;
        langnumber:=Image.Sub(1,6).Getint
    END;
    BEGIN
        REAL ARRAY pc[1:langnumber,1:langnumber];
        TEXT ARRAY rawsub[1:10];
        CLASS language(id,i,n); VALUE id; TEXT id; INTEGER i,n;
        BEGIN  TEXT group; group:-Blanks(n) END;

        REF(language) ARRAY lang[1:langnumber];
        REF(language) temp;
        INTEGER i,j,k,l,m,nm,n,start,fin,minisize,size,maxsize;
        CHARACTER c;
        BOOLEAN assignments;
        REF(Outfile) cooked;

        n:=langnumber;
        FOR i:=1 STEP 1 UNTIL n DO
        BEGIN
            raw.Inimage;
            lang[i]:-NEW language(rawimage.Strip,i,n)
        END;
        FOR i:=1 STEP 1 UNTIL 10 DO
        rawsub[i]:-rawimage.Sub(i*4-3,4);
        Outtext("Reconstructed tree goes to file: ");
        Breakoutimage; Inimage;
        cooked:-NEW Outfile(Sysin.Image.Strip);
        cooked.Open(Blanks(60));
        minisize:=3;

        raw.Inimage; k:=0;
        FOR i:=2 STEP 1 UNTIL n DO
        BEGIN
            l:=i-1;
            FOR j:=1 STEP 1 UNTIL l DO
```

```
    BEGIN
        k:=k+1;
        IF k>10 THEN
        BEGIN
            k:=1; raw.Inimage
        END;
        pc[j,i]:=pc[i,j]:=rawsub[k].Getint/1000
    END
END;
raw.Close;

maxsize:=minisize+1;
assignments:=TRUE;
WHILE maxsize>minisize AND assignments DO
BEGIN
    fin:=maxsize:=0; assignments:=FALSE;
    WHILE n-fin>=minisize DO
    BEGIN
        start:=fin+1; fin:=start+1;
        IF fin<=n THEN
        BEGIN
            WHILE lang[fin].group=lang[fin-1].group DO
            IF fin=n THEN GOTO `exit ELSE fin:=fin+1;
            fin:=fin-1;
            `exit:
            size:=fin-start+1;
            IF size>maxsize THEN maxsize:=size;
            IF size>=minisize THEN
            BEGIN
                REAL ARRAY tr[start:fin,start:fin],s[start:fin];
                INTEGER i,j,k,sum,max,imax,jmax,n;
                REAL mininc,maxinc,increase,xsum,x;
                CHARACTER c;
                BOOLEAN ARRAY assigned[start:fin];
                INTEGER ARRAY group,next[start:fin];

                REAL PROCEDURE score(i,j);
                INTEGER i,j;
                BEGIN
                    REAL sum; INTEGER k,n;
                    k:=j;
                    WHILE i\=0 DO
                    BEGIN
                        WHILE j\=0 DO
                        BEGIN
                            sum:=sum+tr[i,j];
                            n:=n+1; j:=next[j]
                        END;
                        j:=k; i:=next[i]
                    END;
                    score:=sum/n
                END;
```

```
                FOR i:=start+1 STEP 1 UNTIL fin DO
                FOR j:=i-1 STEP -1 UNTIL start DO
                BEGIN
                    tr[j,i]:=tr[i,j]:=x:=pc[lang[i].i,lang[j].i];
                    xsum:=xsum+x
                END;
                k:=fin-start+1;
                maxinc:=mininc:=xsum/(k*(k-1)/2);
                WHILE maxinc>=mininc DO
                BEGIN
                    maxinc:=-99999.0;
                    FOR i:=start+1 STEP 1 UNTIL fin DO
                    IF NOT assigned[i] THEN
                    FOR j:=i-1 STEP -1 UNTIL start DO
                    IF NOT assigned [j] THEN
                    BEGIN
                        increase:=score(i,j);
                        IF increase>maxinc THEN
                             BEGIN
                            imax:=i; jmax:=j; maxinc:=increase
                        END
                    END;
                    IF maxinc>=mininc THEN
                    BEGIN
                        assigned[imax]:=assignments:=TRUE;
                        j:=jmax;
                        WHILE next[j]\=0 DO j:=next[j];
                        next[j]:=imax;
                    END
                END;
                k:=0;
                FOR i:=start STEP 1 UNTIL fin DO
                IF NOT assigned[i] THEN
                BEGIN
                    k:=k+1;
                    c:=Char(k+48);
                    lang[i].group.Putchar(c);
                    j:=next[i];
                    WHILE j\=0 DO
                    BEGIN
                        lang[j].group.Putchar(c);
                        j:=next[j]
                    END
                END
            END;
        END
    END;
    BEGIN
        COMMENT sort, using Shell's algorithm;
        m:=1;
        WHILE m<=n DO m:=m+m;
        m:=(m-1)//2;
```

```
            WHILE m>0 DO
            BEGIN
                nm:=n-m;
                FOR j:=1 STEP 1 UNTIL nm DO
                BEGIN
                    i:=j;
                    l:=i+m;
                    `again:
                    IF lang[l].group<lang[i].group THEN
                    BEGIN
                        temp:-lang[i];
                        lang[i]:-lang[l];
                        lang[l]:-temp;
                        i:=i-m;
                        l:=i+m;
                        IF i>0 THEN GOTO `again
                    END;
                END;
                m:=m//2
            END;
        END
    END;
    BEGIN
        INTEGER i,l,max,width;
        FOR i:=1 STEP 1 UNTIL n DO
        BEGIN
            l:=lang[i].id.Length;
            IF l>max THEN max:=l
        END;
        INSPECT cooked DO FOR i:=1 STEP 1 UNTIL n DO
        INSPECT lang[i] DO
        BEGIN
            width:=Entier(Ln(i)*0.4343+1.0);
            IF width<3 THEN
            Outtext(Blanks(3-width));
            Outtext(Blanks(max-id.Length+1));
            Outtext(id);
            Outtext(" (");
            Outint(i,width);
            Outtext(") ");
            Outtext(group.Strip);
            Outimage
        END;
        cooked.Close
    END
END
END
```

TREE

```
BEGIN
    REF(Infile) raw;
    REF(Outfile) cooked;
    TEXT rawimage,cookedimage;
    INTEGER z,i;
    rawimage:-Blanks(80);
    cookedimage:-Blanks(60);
    Outtext("Table is in file: ");
    Breakoutimage; Inimage;
    raw:-NEW Infile(Sysin.Image.Strip);
    Outtext("Minimum-spanning tree goes to file: ");
    Breakoutimage; Inimage;
    cooked:-NEW Outfile(Sysin.Image.Strip);
    cooked.Open(cookedimage);
    INSPECT raw DO
    BEGIN
        Open(rawimage);
        Inimage;
        z:=Inint;
    END;
    BEGIN
        INTEGER i,j,k,l,n,size,max,imax,jmax;
        TEXT ARRAY rawsub[1:10];
        INTEGER ARRAY t[1:z,1:z],list[1:z];
        BOOLEAN ARRAY linked[1:z];
        TEXT ARRAY langname[1:z];

        n:=z;
        FOR i:=1 STEP 1 UNTIL 10 DO
        rawsub[i]:-rawimage.Sub(i*4-3,4);

        INSPECT raw DO
        BEGIN
            FOR i:=1 STEP 1 UNTIL n DO
            BEGIN
                Inimage;
                langname[i]:-Copy(rawimage.Strip)
            END;
            Inimage;
            FOR i:=2 STEP 1 UNTIL n DO
            BEGIN
                l:=i-1;
                FOR j:=1 STEP 1 UNTIL l DO
                BEGIN
                    k:=k+1;
                    IF k>10 THEN
```

```
                BEGIN
                    k:=1;      Inimage
                END;
                t[i,j]:=t[j,i]:=rawsub[k].Getint
            END
        END;
        Close
    END;
    max:=-99999;
    INSPECT cooked DO
    BEGIN
        FOR i:=2 STEP 1 UNTIL n DO
        FOR j:=i-1 STEP -1 UNTIL 1 DO
        IF t[i,j]>max THEN
        BEGIN
            imax:=i; jmax:=j; max:=t[i,j]
        END;
        size:=2; list[1]:=jmax; list[2]:=imax;
        linked[imax]:=linked[jmax]:=TRUE;
        Outfix(max/1000,3,7); Outchar(' ');
        Outtext(langname[jmax]);
        Outtext(" - ");
        Outtext(langname[imax]);
        Outimage;

        WHILE size<n DO
        BEGIN
            max:=-99999;
            FOR k:=1 STEP 1 UNTIL size DO
            BEGIN
                i:=list[k];
                FOR j:=1 STEP 1 UNTIL n DO
                IF NOT linked[j] THEN
                BEGIN
                    IF t[i,j]>max THEN
                    BEGIN
                        imax:=i; jmax:=j;
                        max:=t[i,j]
                    END
                END
            END;
            size:=size+1; list[size]:=jmax; linked[jmax]:=TRUE;
            Outfix(max/1000,3,7); Outchar(' ');
            Outtext(langname[imax]);
            Outtext(" - ");
            Outtext(langname[jmax]);
            Outimage
        END;
        Close
    END
END
END
```

WILDC

```
Simset BEGIN
    REF(Infile) raw;
    TEXT rawimage;
    TEXT ARRAY rawsub[1:20];
    INTEGER i,k,n,langno,item,itemno,group,groupsize;
    CHARACTER wildcard,null;
    BOOLEAN mergings;
    REF(Outfile) cooked;
    REF(Head) family;

    Link CLASS language(id,n); VALUE id; TEXT id; INTEGER n;
    BEGIN
        TEXT vocabulary;
        BOOLEAN justmerged;
        REF(Head) relatives;
        vocabulary:-Blanks(n);
        relatives:-NEW Head
    END;

    REF(language) lang,lang2,lang3;

    PROCEDURE delete;
    BEGIN
        INTEGER item,itno;
        itno:=itemno;
        FOR item:=1 STEP 1 UNTIL itno DO
        BEGIN
            INTEGER ARRAY fqgloss[1:itno];
            REF(language) ARRAY singleton[1:langno];
            INTEGER maxi,i,fq;
            TEXT v;
            lang:-family.First;
            WHILE lang=/=NONE DO
            BEGIN
                INSPECT lang DO
                BEGIN
                    vocabulary.Setpos(item);
                    i:=Rank(vocabulary.Getchar);
                    IF i=1 AND justmerged THEN
                    BEGIN
                        vocabulary.Setpos(vocabulary.Pos-1);
                        vocabulary.Putchar(null)
                    END;
                    IF i>maxi THEN maxi:=i;
                    IF i>1 THEN
                    BEGIN
```

```
                    fq:=fqgloss[i];
                    IF fq=0 THEN singleton[i]:-lang;
                    fqgloss[i]:=fq+1
                END
            END;
            lang:-lang.Suc
        END;
        FOR i:=2 STEP 1 UNTIL maxi DO
        IF fqgloss[i]=1 THEN INSPECT singleton[i] DO
        BEGIN
            vocabulary.Setpos(item);
            vocabulary.Putchar(wildcard)
        END
    END;
END;

PROCEDURE innovated(v,l,n);
REF(language) l; INTEGER n; TEXT v;
INSPECT l DO
INSPECT cooked DO
BEGIN
    INTEGER ARRAY inno[1:n];
    INTEGER i,k;
    CHARACTER vc,lc;
    vocabulary.Setpos(1); v.Setpos(1);
    Outtext(id); Outtext(" replaced ");
    WHILE vocabulary.More DO
    BEGIN
        vc:=v.Getchar; lc:=vocabulary.Getchar;
        IF lc=wildcard AND vc\=null THEN
        BEGIN
            k:=k+1; inno[k]:=vocabulary.Pos-1
        END
    END;
    Outint(k,IF k=0 THEN 1 ELSE Entier(Ln(k)*0.4343+1.0));
    Outtext (" item");
    IF k\=1 THEN Outchar('s');
    Outchar(':');
    FOR i:=1 STEP 1 UNTIL k DO Outint(inno[i],3);
    IF Image.Strip=/=NOTEXT THEN Outimage;
    vocabulary.Setpos(1);
    WHILE vocabulary.More DO
    BEGIN
        k:=Rank(vocabulary.Getchar);
        Outchar(
        IF k=0 THEN '.' ELSE IF k=1 THEN '*'
        ELSE Char(IF k<11 THEN k+46
        ELSE IF k<37 THEN k+54 ELSE k+60))
    END;
    IF Image.Strip=/=NOTEXT THEN Outimage;
END;
```

```
Outtext("Wordlists (reduced to cognate groups) are in file: ");
Breakoutimage; Inimage;
raw:-NEW Infile(Sysin.Image.Strip);
rawimage:-Blanks(60);
INSPECT raw DO
BEGIN
    Open(rawimage);
    Inimage;
    langno:=Image.Sub(1,6).Getint;
    itemno:=Image.Sub(7,6).Getint
END;
wildcard:=Char(1); null:=Char(0);
groupsize:=1;

Outtext("Reconstruction goes to file: ");
Breakoutimage; Inimage;
cooked:-NEW Outfile(Sysin.Image.Strip);
cooked.Open(Blanks(50));
family:-NEW Head;
FOR i:=1 STEP 1 UNTIL langno DO
BEGIN
    raw.Inimage;
    NEW language(rawimage.Strip,itemno).Into(family);
END;
FOR i:=1 STEP 1 UNTIL 20 DO
rawsub[i]:-rawimage.Sub(i*3-2,3);
lang:-family.First;
WHILE lang=/=NONE DO
BEGIN
    raw.Inimage; k:=1;
    INSPECT lang DO
    FOR item:=1 STEP 1 UNTIL itemno DO
    BEGIN
        vocabulary.Putchar(Char(rawsub[k].Getint+2));
        IF k=30 THEN
        BEGIN
            raw.Inimage; k:=1
        END
        ELSE k:=k+1
    END;
    lang:-lang.Suc
END;
delete;
mergings:=TRUE;
WHILE family.Cardinal>1  AND mergings DO
BEGIN
    mergings:=FALSE;
    lang:-family.First;
    WHILE lang=/=family.Last AND lang=/=NONE DO
    BEGIN
        TEXT v1,v2;
        INTEGER n,sum;
```

```
        REAL maxr;
        CHARACTER c1,c2;
        v1:-lang.vocabulary;
        lang2:-lang.Suc;
        WHILE lang2=/=NONE DO
        BEGIN
            v2:-lang2.vocabulary;
            v1.Setpos(1); v2.Setpos(1);
            WHILE v1.More DO
            BEGIN
                c1:=v1.Getchar; c2:=v2.Getchar;
                IF c1<=wildcard OR c2<=wildcard THEN
                BEGIN
                    sum:=sum+1; n:=n+1
                END
                ELSE
                BEGIN
                    n:=n+1; IF c1=c2 THEN sum:=sum+1
                END
            END;
            lang3:-lang2;
            lang2:-lang2.Suc;
            IF sum=n THEN
            BEGIN
                lang3.Into(lang.relatives);
                mergings:=TRUE
            END
        END;
        maxr:=0.0; lang:-lang.Suc
    END;
    IF mergings THEN
    BEGIN
        lang:-family.First;
        WHILE lang=/=NONE DO
        BEGIN
            TEXT newvocabulary,v2; CHARACTER c1,c2;
            INSPECT lang DO
            INSPECT relatives DO
            IF Empty THEN justmerged:=FALSE ELSE
            BEGIN
                TEXT t,newid;
                group:=group+1;
                justmerged:=TRUE;
                IF Mod(group,10)=0 THEN groupsize:=groupsize+1;
                cooked.Outtext("Group#");
                cooked.Outint(group,groupsize);
                cooked.Outtext(" = ");
                cooked.Outtext(id);
                newvocabulary:-Copy(vocabulary);
                t:-Copy("Group#    ");
                t.Sub(7,groupsize).Putint(group);
                newid:-Copy(t.Strip);
```

```
            lang2:-First;
            WHILE lang2=/=NONE DO
            BEGIN
                cooked.Outtext(", ");
                cooked.Outtext(lang2.id);
                v2:-lang2.vocabulary;
                newvocabulary.Setpos(1); v2.Setpos(1);
                WHILE newvocabulary.More DO
                BEGIN
                    c1:=newvocabulary.Getchar;
                    c2:=v2.Getchar;
                    IF c1<=wildcard AND c1<=c2 THEN
                    BEGIN
                        newvocabulary.Setpos
                        (newvocabulary.Pos-1);
                        newvocabulary.Putchar
                        (IF c2=wildcard THEN null ELSE c2)
                    END
                END;
                lang2:-lang2.Suc
            END;
            INSPECT cooked DO
            BEGIN
                Outimage;
                newvocabulary.Setpos(1);
                WHILE newvocabulary.More DO
                BEGIN
                    k:=Rank(newvocabulary.Getchar);
                    Outchar(IF k=1 THEN '*'
                    ELSE Char(IF k<11 THEN k+46
                    ELSE IF k<37 THEN k+54 ELSE k+60))
                END;
                IF Image.Strip=/=NOTEXT THEN Outimage;
            END;
            lang2:-First;
            innovated(lang2.vocabulary,lang,itemno);
            WHILE lang2=/=NONE DO
            BEGIN
                innovated(lang.vocabulary,lang2,itemno);
                lang2:-lang2.Suc
            END;
            lang.id:-Copy(newid);
            lang.vocabulary:=newvocabulary;
            cooked.Outimage; Clear
        END;
        lang:-lang.Suc
    END;
    delete
END
```

```
        ELSE
        BEGIN
            cooked.Outtext
            ("??? Misidentified cognates, I cannot continue.");
            cooked.Outimage
        END
    END;
    cooked.Close; raw.Close
END
```

REFERENCES

DYEN, Isidore, A.T. JAMES and J.W.L. COLE
1967 Language Divergence and Estimated Word Retention Rate. Language 43/1:150-171.

FODOR, Istvan
1965 The Rate of Linguistic Change. The Hague: Mouton.

GUDSCHINSKY, Sarah C.
1956 The ABC's of Lexicostatistics. Word 12/2:175-210.

GUY, Jacques B.M.
1978 Proto-North New Hebridean Reconstructions. Pacific Linguistics C-61:781-850.

HARTIGAN, John A.
1974 Clustering Algorithms. New York: Wiley.

HATTORI, Shiro
1954 On the Method of Glottochronology and the Time-depth of Proto-Japanese. Gengo Kenkyu 26:29-77.
1962 Comments to Knut Bergsland and Hans Vogt "On the Validity of Glottochronology". Current Anthropology 3/2:134-135.

ROSS, Malcolm
1979 The Austronesian Languages of Papua: Towards a Family Tree. TS.

SANDERS, Arden G.
1977 Some Synchronic Analysis Procedures for Language Survey Data. Workpapers in Papua New Guinea Languages 21:295-315.

SIMONS, Gary
1977 Tables of Significance for Lexicostatistics. Workpapers in Papua New Guinea Languages 21:75-106.

SPIEGEL, Murray R.
1972 Schaum's Outline of Theory and Problems of Statistics in SI Units. New York: McGraw-Hill.

SUKHOTIN, B.V.
1962 Eksperimental'noe vydelenie klassov bukv s pomoshchju elektronnoj vychislitel'noj mashiny. Problemy strukturnoy lingvistiki 234:198-206.

TRYON, D.T.
1973 Linguistic Subgrouping in the New Hebrides: A Preliminary Approach. Oceanic Linguistics 12/1-2:303-351.
1976 New Hebrides Languages: an Internal Classification. Pacific Linguistics C-50.
1978 The Languages of the New Hebrides. Pacific Linguistics C-61:877-897.

Guy, J.B.M. *Experimental glottochronology: Basic methods and results.*
B-75, viii + 224 pages. Pacific Linguistics, The Australian National University, 1980. DOI:*10.15144/PL-B75.1*