April 12th
clustering for historical linguistics
An IndoEuropean Classification:
A Lexicostatistical Experiment

ISIDORE DYEN
JOSEPH B. KRUSKAL
PAUL BLACK

b. 8/16/1913
d. 12/14/2008

http://www.wordgumbo.com/ie/cmp/iedata.txt
Morris Swadesh’s concepts

a 001 ALL
a 002 AND
a 003 ANIMAL
a 004 ASHES
a 005 AT
a 006 BACK
a 007 BAD
a 008 BARK (OF A TREE)
a 009 BECAUSE
a 010 BELLY
a 011 BIG
a 012 BIRD
a 013 TO BITE
a 014 BLACK
a 015 BLOOD
a 016 TO BLOW
(a 017 BONE)
a 018 TO BREATHE
(a 019 TO BURN)
(INTRANSITIVE)
a 020 CHILD (YOUNG)
a 021 CLOUD
(a 022 COLD)
(WEATHER)
a 023 TO COME
a 024 TO COUNT
a 025 TO CUT
a 026 DAY (NOT NIGHT)
a 027 TO DIE
a 028 TO DIG
a 029 DIRTY
a 030 DOG
a 031 TO DRINK
(a 032 DRY)
a 033 DULL (KNIFE)
a 034 DUST
a 035 EAR
a 036 EARTH (SOIL)
a 037 TO EAT
a 038 EGG
(a 039 EYE)
a 040 TO FALL
(DROP)
a 041 FAR
(a 042 FAT)
a 043 FATHER
(a 044 TO FEAR)
a 045 FEATHER
(a 046 FEW)
a 047 TO FIGHT
(a 048 FIRE)
(GAME)
a 049 FISH
a 050 FIVE
(a 051 TO FLOAT)
a 052 TO FLOW
a 053 FLOWER
a 054 TO FLY
a 055 FOG
a 056 FOOT
a 057 FOUR
(a 058 TO FREEZE)
a 059 FRUIT
a 060 TO GIVE
(a 061 GOOD)
a 062 GRASS
(a 063 GREEN)
a 064 GUTS
(a 065 HAIR)
a 066 HAND
(a 067 HE)
a 068 HEAD
(a 069 TO HEAR)
a 070 HEART
(a 071 HEAVY)
a 072 HERE
(a 073 TO HIT)
a 074 HOLD (IN HAND)
(a 075 HOW)
a 076 TO HUNT
(GAME)
a 077 HUSBAND
a 078 I
(a 079 ICE)
a 080 IF
(a 081 IN)
a 082 TO KILL
(a 083 KNOW (FACTS))
a 084 LAKE
(a 085 TO LAUGH)
a 086 LEAF
(a 087 LEFT (HAND))
a 088 LEG
(a 089 TO LIE (ON SIDE))
a 090 TO LIVE
a 091 LIVER
(a 092 LONG)
a 093 LOUSE
(a 094 MAN (MALE))
a 095 MANY
a 096 MEAT (FLESH)
a 097 MOTHER
(a 098 MOUNTAIN)
a 099 MOUTH
(a 100 NAME)
a 101 NARROW
(a 102 NEAR)
a 103 NECK
(a 104 NEW)
a 105 NIGHT
(a 106 NOSE)
a 107 NOT
The speech forms from these lists were then punched onto IBM cards using the limited set of characters available on cards at that time, yielding one "form card" for each Swadesh meaning in each list.

... 

For each meaning, the forms on the form cards were examined and cognation judgments made by Isidore Dyen. (Note that cognation judgments are only made between forms having the same meaning. This is an important aspect of the lexicostatistical method.)
Concept #11, BIG

<table>
<thead>
<tr>
<th>Language</th>
<th>Code</th>
<th>Language</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Russian</td>
<td>BOL SOJ</td>
<td>48 Ukrainian</td>
<td>VELYKYJ</td>
</tr>
<tr>
<td>Swedish List</td>
<td>GROV</td>
<td>49 Byelorussian</td>
<td>VJALIKI</td>
</tr>
<tr>
<td>Latvian</td>
<td>LIELS</td>
<td>47 Czech E</td>
<td>VELIKE, HRUBE</td>
</tr>
<tr>
<td>Afghan</td>
<td>LOJ</td>
<td>58 Marathi</td>
<td>MOTHA</td>
</tr>
<tr>
<td>Singhalese</td>
<td>LOKU</td>
<td>59 Gujarati</td>
<td>MOTU</td>
</tr>
<tr>
<td>Wakhi</td>
<td>LUP</td>
<td>04 Welsh C</td>
<td>MAWR</td>
</tr>
<tr>
<td>60 Panjabi ST</td>
<td>VEDDA</td>
<td>03 Welsh N</td>
<td>MAWR</td>
</tr>
<tr>
<td>61 Lahnda</td>
<td>WEDDA</td>
<td>01 Irish A</td>
<td>MOR</td>
</tr>
<tr>
<td>53 Bulgarian</td>
<td>GOLJAMO</td>
<td>02 Irish B</td>
<td>MOR</td>
</tr>
<tr>
<td>52 Macedonian</td>
<td>GOLEM</td>
<td>24 German ST</td>
<td>GROSS</td>
</tr>
<tr>
<td>94 BULGARIAN P</td>
<td>GOL AM</td>
<td>27 Afrikaans</td>
<td>GROOT</td>
</tr>
<tr>
<td>77 Tadzik</td>
<td>KALON, BUZURG</td>
<td>26 Dutch List</td>
<td>GROOT</td>
</tr>
<tr>
<td>76 Persian List</td>
<td>BOZORG</td>
<td>25 Penn. Dutch</td>
<td>GROESZ</td>
</tr>
<tr>
<td>40 Lithuanian ST</td>
<td>DIDELOS</td>
<td>28 Flemish</td>
<td>GROOT</td>
</tr>
<tr>
<td>39 Lithuanian O</td>
<td>DIDELOS</td>
<td>29 Frisian</td>
<td>GREAT</td>
</tr>
<tr>
<td>85 RUSSIAN P</td>
<td>VELIKIJ</td>
<td>08 Rumanian List</td>
<td>MARE</td>
</tr>
<tr>
<td>54 Serbocroatian</td>
<td>VELIK</td>
<td>09 Vlach</td>
<td>MARE</td>
</tr>
<tr>
<td>92 SERBOCROATIAN P</td>
<td>VELIK</td>
<td>38 Takitaki</td>
<td>BIGI, LANGA,</td>
</tr>
<tr>
<td>46 Slovak</td>
<td>VEL KY</td>
<td>37 English ST</td>
<td>BIG</td>
</tr>
<tr>
<td>89 SLOVAK P</td>
<td>VEL KY</td>
<td>78 Baluchi</td>
<td>MAZAAN, MAZ EN</td>
</tr>
<tr>
<td>42 Slovenian</td>
<td>VELIKA</td>
<td>17 Sardinian N</td>
<td>MANNU</td>
</tr>
<tr>
<td>91 SLOVENIAN P</td>
<td>VELIK</td>
<td>18 Sardinian L</td>
<td>MANNU</td>
</tr>
<tr>
<td>86 UKRAINIAN P</td>
<td>VELIKYJ</td>
<td>19 Sardinian C</td>
<td>MANNU</td>
</tr>
<tr>
<td>87 BYELORUSSIAN P</td>
<td>V ALIKI</td>
<td>68 Greek Mod</td>
<td>MEGHALOS</td>
</tr>
<tr>
<td>45 Czech</td>
<td>VELKY</td>
<td>66 Greek ML</td>
<td>MEGALOS</td>
</tr>
<tr>
<td>90 CZECH P</td>
<td>VELKY</td>
<td>70 Greek K</td>
<td>MEGALOS</td>
</tr>
<tr>
<td>43 Lusatian L</td>
<td>WELIKI</td>
<td>67 Greek MD</td>
<td>MEGALOS</td>
</tr>
<tr>
<td>44 Lusatian U</td>
<td>WULKI</td>
<td>69 Greek D</td>
<td>MEGALOS</td>
</tr>
<tr>
<td>93 MACEDONIAN P</td>
<td>VELIK</td>
<td>71 Armenian Mod</td>
<td>MEC</td>
</tr>
<tr>
<td>50 Polish</td>
<td>WIELKI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>88 POLISH P</td>
<td>WIELKI</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Monday, April 12, 2010
fraction of cognates for Swadesh concepts

IE-perc84.txt

Monday, April 12, 2010
m <- read.table("IE-perc84.txt")
Hierarchical Clustering

Description
Hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it.

Usage
hclust(d, method = "complete", members=NULL)

## S3 method for class 'hclust':
plot(x, labels = NULL, hang = 0.1,
     axes = TRUE, frame.plot = FALSE, ann = TRUE,
     main = "Cluster Dendrogram",
     sub = NULL, xlab = NULL, ylab = "Height", ...)

plclust(tree, hang = 0.1, unit = FALSE, level = FALSE, hmin = 0,
        square = TRUE, labels = NULL, plot. = TRUE,
        axes = TRUE, frame.plot = FALSE, ann = TRUE,
        main = ",", sub = NULL, xlab = NULL, ylab = "Height")

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>a dissimilarity structure as produced by dist.</td>
</tr>
</tbody>
</table>
m <- read.table("IE-perc84.txt")
# change from a fraction out of 1000 to a Percent Cognate
subtr <- function (x) c((1000-x)/10)
# apply this function to all cells of the cognate matrix
d <- apply(m[1:84],c(1,2),subtr)

# set the column names to be meaningful
colnames(d) <- m$V85
# do the clustering
clus <- hclust(as.dist(d), "ave")
# show it
plot(clus,cex=0.5)
```r
m <- read.table("IE-perc84.txt")
# change from a fraction out of 1000 to a Percent Cognate
substr <- function (x) c((1000-x)/10)
# apply this function to all cells of the cognate matrix
d <- apply(m[1:84],c(1,2),substr)
# set the column names to be meaningful
colnames(d) <- m$V85
# do the clustering
clus <- hclust(as.dist(d), "ave")
# show it
plot(clus,cex=0.5)
```
1. Find the two most similar items
2. Link them together into a cluster.
3. Repeat until all items are in a cluster
Basis on which to decide similarity
Single linkage

\[ \min\{ d(x, y) : x \in A, y \in B \} \]
Complete linkage

\[
\max \{ d(x, y) : x \in A, y \in B \}
\]
Average linkage

\[ \frac{1}{|A| \cdot |B|} \sum_{x \in A} \sum_{y \in B} d(x, y) \]
Resort to spelling

<table>
<thead>
<tr>
<th>I N T E N T I O N</th>
</tr>
</thead>
<tbody>
<tr>
<td>* E X E C U T I O N</td>
</tr>
<tr>
<td>d s s i s</td>
</tr>
</tbody>
</table>

**Figure 3.23** Representing the minimum edit distance between two strings as an **alignment**. The final row gives the operation list for converting the top string into the bottom string: d for deletion, s for substitution, i for insertion.
intention
intention ← delete i
attention ← substitute n by e
exention ← substitute t by x
execution ← insert u
execution ← substitute n by c

Figure 3.24 Transformation list example of Kruskal (1983) from intention to execution.
Levenshtein distance
aka “dynamic time warping”

\[
\text{distance}[i, j] = \min \left\{ \begin{array}{l}
\text{distance}[i - 1, j] + \text{ins-cost}(\text{target}_{i-1}) \\
\text{distance}[i - 1, j - 1] + \text{sub-cost}(\text{source}_{j-1}, \text{target}_{i-1}) \\
\text{distance}[i, j - 1] + \text{del-cost}(\text{source}_{j-1}) \\
\end{array} \right. 
\]
Johnson's price list

\[ \text{distance}[i, j] = \min \left\{ \begin{array}{l}
\text{distance}[i-1, j] + \text{ins-cost}(\text{target}_{i-1}) \\
\text{distance}[i-1, j-1] + \text{sub-cost}(\text{source}_{j-1}, \text{target}_{i-1}) \\
\text{distance}[i, j-1] + \text{del-cost}(\text{source}_{j-1})
\end{array} \right\} \]

\begin{align*}
\text{sub-cost:vowel} & \quad \$1.00 \\
\text{sub-cost:const} & \quad \$2.00 \\
\text{ins-cost} & \quad \$1.50 \\
\text{deletion} & \quad \$1.50
\end{align*}
**DTW algorithm**

```plaintext
function MIN-EDIT-DISTANCE(target, source) returns min-distance

n ← LENGTH(target)
m ← LENGTH(source)
Create a distance matrix distance[n+1,m+1]
Initialize the zeroth row and column to be the distance from the empty string
distance[0,0] = 0
for each column i from 1 to n do
    distance[i,0] ← distance[i-1,0] + ins-cost(target[i])
for each row j from 1 to m do
    distance[0,j] ← distance[0,j-1] + del-cost(source[j])
for each column i from 1 to n do
    for each row j from 1 to m do
        distance[i,j] ← MIN( distance[i-1,j] + ins-cost(target[i-1]),
                               distance[i-1,j-1] + sub-cost(source[j-1], target[i-1]),
                               distance[i,j-1] + del-cost(source[j-1]))

return distance[n,m]
```

**Figure 3.25** The minimum edit distance algorithm, an example of the class of dynamic programming algorithms. The various costs can either be fixed (e.g., $\forall x, \text{ins-cost}(x) = 1$) or can be specific to the letter (to model the fact that some letters are more likely to be inserted than others). We assume that there is no cost for substituting a letter for itself (i.e., $\text{sub-cost}(x,x) = 0$).
The minimum edit distance algorithm, an example of the class of dynamic programming algorithms. The various costs can either be fixed (e.g., \( \forall x, \text{ins-cost}(x) = 1 \)) or can be specific to the letter (to model the fact that some letters are more likely to be inserted than others). We assume that there is no cost for substituting a letter for itself (i.e., \( \text{sub-cost}(x, x) = 0 \)).

Keith divides this by \(|n|+|m|\)
Figure 3.27 When entering a value in each cell, we mark which of the three neighboring cells we came from with up to three arrows. After the table is full we compute an alignment (minimum edit path) by using a backtrace, starting at the 8 in the upper-right corner and following the arrows. The sequence of dark grey cells represents one possible minimum cost alignment between the two strings.
Lahnda vs Panjabi

1;Lahnda;sebu,sare 1;Panjabi ST;seb,sara
2;Lahnda;te 2;Panjabi ST;te
3;Lahnda;janwer 3;Panjabi ST;janver
4;Lahnda;swa 4;Panjabi ST;sva
5;Lahnda;ute 5;Panjabi ST;te,per
6;Lahnda;picche 6;Panjabi ST;pitth
7;Lahnda;bura 7;Panjabi ST;pera
8;Lahnda;chal 8;Panjabi ST;chell
9;Lahnda;kyuke 9;Panjabi ST;kyoke
10;Lahnda;pet 10;Panjabi ST;pet
11;Lahnda;wedda 11;Panjabi ST;vedda
12;Lahnda;perinda 12;Panjabi ST;penchi
13;Lahnda;ketten 13;Panjabi ST;veddena
14;Lahnda;kala 14;Panjabi ST;kala
15;Lahnda;xun 15;Panjabi ST;leu
16;Lahnda;cellen 16;Panjabi ST;vegna
17;Lahnda;heddi 17;Panjabi ST;heddi
18;Lahnda;sa ghinnen 18;Panjabi ST;salena
19;Lahnda;bha lawen 19;Panjabi ST;jelna
20;Lahnda;becca 20;Panjabi ST;becca
21;Lahnda;beddel 21;Panjabi ST;beddel
22;Lahnda;thend 22;Panjabi ST;thend
23;Lahnda;awen 23;Panjabi ST;ona

...lots of cognates

IE-lists.txt
print STDERR "calculating distance for word number (out of 200): ";
foreach $wnum (sort {$a <=> $b} keys %wtable) { # loop over the 200 words
  # testing - loop over a subset of words
  print STDERR "$wnum ";
  foreach $l1 (sort keys %{wtable->{$wnum}}) { # loop over the languages
    foreach $l2 (sort keys %{wtable->{$wnum}}) { # again
      if ($l1 eq $l2) { # lang against itself is distance of 0
        Stotald{$l1}{$l2}=0;
        Scount{$l1}{$l2}=1;
        next;
      }
      if ($l2 le $l1) {next;}; # look at each language pair only once
      $min = 100;
      for ($i=0; $i<@{wtable->{$wnum}{$l1}}; $i++) { # each form in l1
        for ($j=0; $j<@{wtable->{$wnum}{$l2}}; $j++) { # and each in l2
          # leven() is the DTW distance algorithm
          $d = leven(wtable->{$wnum}{$l1}[$i], wtable->{$wnum}{$l2}[$j]);
          if ($d < $min) {
            $min=$d;
            print "$wnum, $l1, $l2, cite, actual, $d\n";
          }
        }
      }
      Scount{$l1}{$l2}++; # the total is based on "count" words
      # add the min distance between two langs for this word
      Stotald{$l1}{$l2}+= $min;
    }
  }
}

foreach $l1 (sort keys %stotald) {
  foreach $l2 (sort keys %{stotald->{$l1}}) {
    $save = stotald{$l1}{$l2}/scount{$l1}{$l2};
    print "$l1, $l2, Stotald{$l1}{$l2}, Scount{$l1}{$l2}, $save\n";
  }
}

get_IE_distance:42
Spelling similarities between Lahnda and Panjabi

distance between Lahnda and itself is 0

0.740, 0.773, 0.807, 0.790, 0.775, 0.790, 0.782, 0.789, 0.802, 0.809, 0.698, 0.561, 0.776, 0.772, 0.751, 0.776, 0.796, 0.812, 0.766, 0.781, 0.789, 0.815, 0.742, 0.820, 0.812, 0.733, 0.784, 0.792, 0.807, 0.782, 0.752, 0.793, 0.800, 0.786, 0.801, 0.815, 0.517, 0.691, 0.390, 0.816, 0.856, 0.835, 0.764, 0.594, 0.590, 0.780, 0.000, 0.806, 0.817, 0.797, 0.799, 0.796, 0.792, 0.767, 0.562, 0.516, 0.754, 0.328, 0.837, 0.696, 0.813, 0.770, 0.761, 0.824, 0.791, 0.799, 0.736, 0.781, 0.782, 0.769, 0.797, 0.727, 0.795, 0.791, 0.775, 0.817, 0.818, 0.833, 0.667, 0.797, 0.791, 0.789, 0.739, 0.779, 0.733, 0.837, 0.829, Lahnda

distance between Lahnda and Panjabi is 0.328
-- very low compared to these other 0.7, 0.8s etc

IE-distmatrix.txt:47
hclust (*, "complete")
as.dist(d)
Phonetic similarity

Compare Two Phones

agree in sonorance?

YES

both +SON or -SON?

-SON

agree in place?

YES

agree in manner?

YES

agree in voicing?

YES

0.0

NO

0.25

agree in voicing?

NO

0.4

0.6

+SON

agree as being both r/l, vowels or nasals?

YES

agree in place?

YES

0.75

agree in manner?

NO

0.75

agree in voicing?

YES

0.0

NO

0.25

agree in place?

NO

1.0

Sarah Brooks
WashU undergrad honors thesis

Monday, April 12, 2010
Exercise

Try cluster analysis yourself! Do 1,2, or 3 on p 214-215 of Johnson