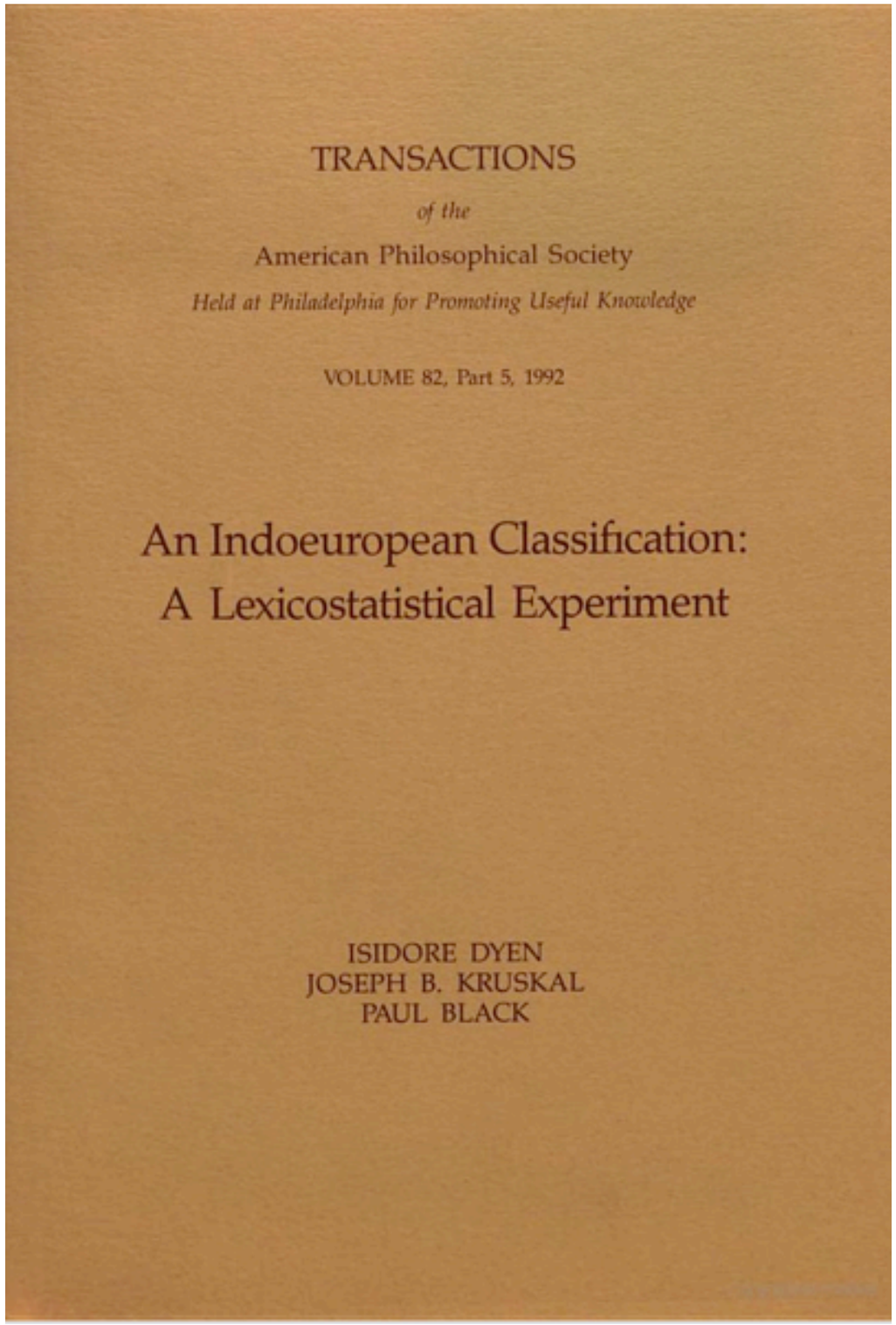
April 12th
clustering for historical linguistics

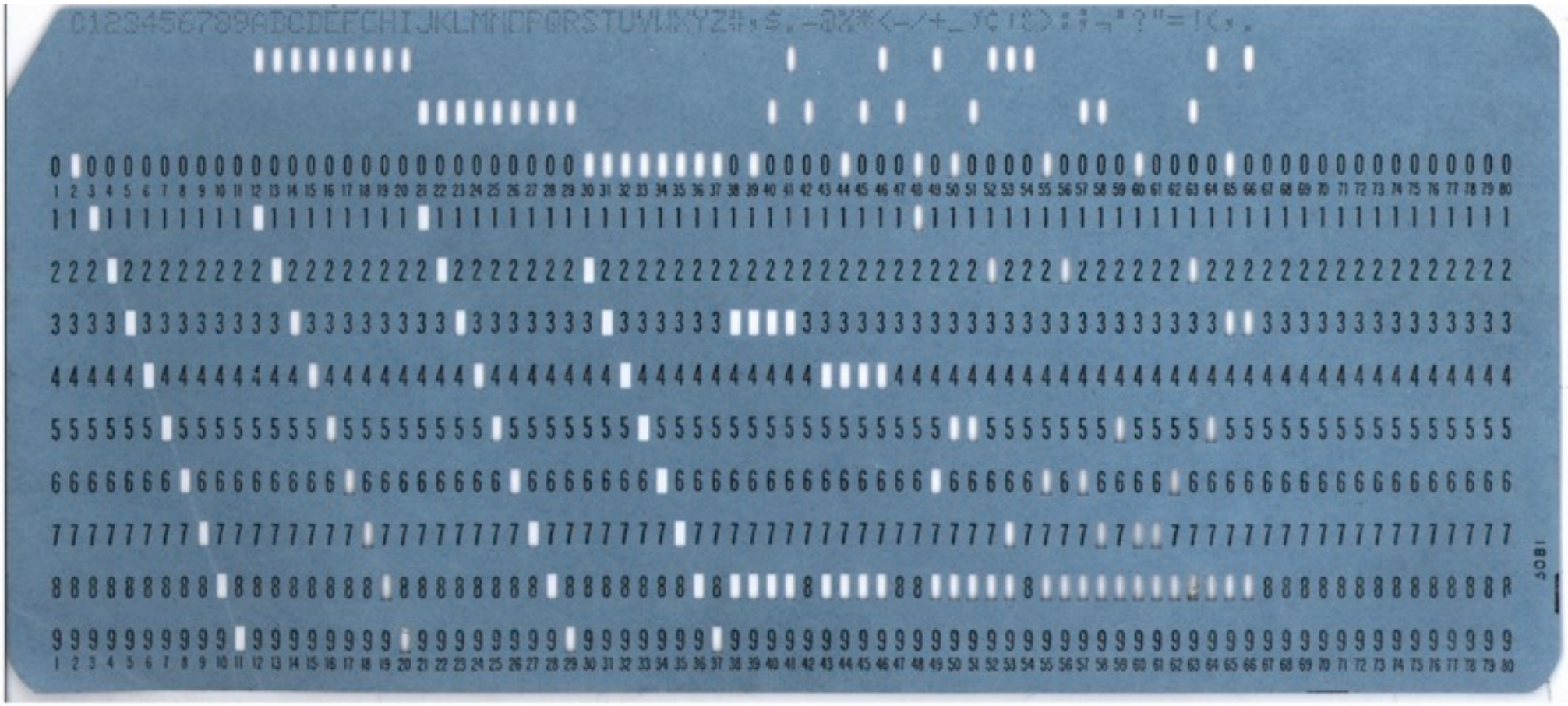


b. 8/16/1913
d. 12/14/2008

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 (WIND)
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 (SUBSTANCE)
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 (SUBSTANCE)
a 043 FATHER
a 044 TO FEAR
a 045 FEATHER (LARGE)
a 046 FEW
a 047 TO FIGHT
a 048 FIRE
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

51	Russian	BOL SOJ
32	Swedish List	GROV
41	Latvian	LIELS
74	Afghan	LOJ
56	Singhalese	LOKU
79	Wakhi	LUP
60	Panjabi ST	VEDDA
61	Lahnda	WEDDA
53	Bulgarian	GOLJAMO
52	Macedonian	GOLEM
94	BULGARIAN P	GOL AM
77	Tadzik	KALON, BUZURG
76	Persian List	BOZORG
40	Lithuanian ST	DIDELIS
39	Lithuanian O	DIDELIS
85	RUSSIAN P	VELIKIJ
54	Serbocroatian	VELIK
92	SERBOCROATIAN P	VELIK
46	Slovak	VEL KY
89	SLOVAK P	VEL KY
42	Slovenian	VELIKA
91	SLOVENIAN P	VELIK
86	UKRAINIAN P	VELIKYJ
87	BYELORUSSIAN P	V ALIKI
45	Czech	VELKY
90	CZECH P	VELKY
43	Lusatian L	WELIKI
44	Lusatian U	WULKI
93	MACEDONIAN P	VELIK
50	Polish	WIELKI
88	POLISH P	WIELKI

48	Ukrainian	VELYKYJ, SLJAXETNYJ
49	Byelorussian	VJALIKI
47	Czech E	VELIKE, HRUBE
58	Marathi	MOTHA
59	Gujarati	MOTU
04	Welsh C	MAWR
03	Welsh N	MAWR
01	Irish A	MOR
02	Irish B	MOR
24	German ST	GROSS
27	Afrikaans	GROOT
26	Dutch List	GROOT
25	Penn. Dutch	GROESZ
28	Flemish	GROOT
29	Frisian	GREAT
08	Rumanian List	MARE
09	Vlach	MARE
38	Takitaki	BIGI, LANGA, GROFOE, GRAN, GRANDI
37	English ST	BIG
78	Baluchi	MAZAAN, MAZ EN
17	Sardinian N	MANNU
18	Sardinian L	MANNU
19	Sardinian C	MANNU
68	Greek Mod	MEGHALOS
66	Greek ML	MEGALOS
70	Greek K	MEGALOS
67	Greek MD	MEGALOS
69	Greek D	MEGALOS
71	Armenian Mod	MEC

fraction of cognates for Swadesh concepts

```
0 826 344 342 311 309 319 149 145 179 165 168 163 168 123 124 155 168 154 168 163 166 155 179 164 170 169 176 165 189 190 191 189 170 194 180 179 156 191 189 170 180 173 178 174 173 162 181 173 173 192 200 171 182 101 102 118 118 111 119 119 119 119 128 115 133 151 142 144 159 120 112 111  93  91 104 120 100  82  95  99 104  98 109 Irish_A
826   0 355 360 335 321 328 163 160 200 179 193 188 193 147 150 175 188 180 195 183 193 176 194 171 180 179 181 175 184 184 186 183 164 184 170 183 160 198 196 176 191 206 211 212 205 189 214 200 200 218 207 182 204 109  99 121 121 114 117 122 122 122 131 118 131 150 141 142 159 122 109 103 105  93 112 129 109  90  98 107 106 101 111 Irish_B
344 355   0 939 639 629 641 188 189 207 199 195 190 195 176 174 194 207 194 187 196 200 182 180 156 160 161 168 167 165 166 181 175 151 180 173 159 160 189 193 158 162 185 185 200 194 185 188 184 178 182 187 162 179  97  87 129 107 101 115 114 124 119 115 122 124 137 133 135 155 126 113 101  93  82 101 116  96 100  94  98  99  94 104 Welsh_N
342 360 939   0 615 610 617 193 193 211 203 199 194 199 175 173 198 211 197 190 200 203 187 185 160 164 165 172 171 169 170 186 179 150 184 177 163 164 188 186 156 171 188 188 203 197 188 192 188 181 186 190 166 182  96  92 129 112 105 119 119 129 124 119 122 128 141 138 140 160 131 118 106  92  87 105 121 101 105  94  98  98  94 104 Welsh_C
311 335 639 615   0 888 929 212 202 236 218 224 214 219 201 199 223 236 222 219 219 223 207 206 187 186 191 188 188 171 177 193 191 172 185 183 169 164 182 191 155 182 201 201 211 199 197 194 196 188 198 202 177 199 101  97 134 118 105 125 124 135 130 130 139 134 147 144 146 166 126 112 102  98  92 105 121 100 105 105 108 110 104 115 Breton_List
309 321 629 610 888   0 949 212 209 226 212 214 204 209 191 188 212 226 211 209 214 218 202 205 187 185 190 188 194 174 180 196 195 176 189 188 179 176 187 190 160 181 199 199 209 198 195 193 194 188 196 207 181 198 107 102 138 122 110 130 130 140 135 135 139 129 141 138 141 160 126 112 112 103  97 116 126 111 118 109 113 114 109 120 Breton_SE
319 328 641 617 929 949   0 214 210 232 219 221 210 215 197 195 219 232 218 215 221 224 208 206 188 186 191 188 194 180 187 202 196 177 196 194 179 176 188 192 161 182 200 201 211 199 196 194 195 188 197 209 182 199 108 103 140 123 110 131 130 141 135 135 140 130 142 139 141 166 126 112 118 104  98 116 127 111 118 109 114 115 110 121 Breton_ST
149 163 188 193 212 212 214   0 723 660 643 622 579 584 513 513 577 602 598 594 629 608 577 249 234 254 249 232 229 222 218 239 237 214 223 216 227 212 199 203 179 210 230 230 223 227 214 201 212 216 219 222 202 222 134 136 161 162 141 173 168 173 168 168 147 147 150 157 154 168 136 118 138  94  98 136 137 132 149 109 113 115 113 125 Rumanian_List
145 160 189 193 202 209 210 723   0 592 562 562 511 525 472 483 528 539 542 531 547 537 540 247 229 249 247 244 240 236 232 236 236 222 229 218 240 211 219 218 202 236 244 244 233 237 229 220 226 237 236 240 225 236 127 144 171 166 144 172 168 172 167 173 165 178 177 183 180 191 149 129 139 103 107 149 157 138 150 119 124 124 124 129 Vlach
179 200 207 211 236 226 232 660 592   0 832 818 803 774 697 689 728 797 731 788 773 785 764 265 250 260 259 251 245 254 250 259 263 246 245 242 247 215 234 242 218 240 250 250 247 251 236 226 237 236 239 250 231 245 133 152 173 167 152 184 183 182 173 178 168 168 177 178 174 190 152 138 143 120 114 141 157 136 140 115 123 120 122 134 Italian
165 179 199 203 218 212 219 643 562 832   0 760 740 731 653 660 655 687 662 716 731 722 696 250 241 246 246 234 231 240 236 251 249 227 230 224 239 212 219 218 209 227 251 251 240 249 234 223 234 233 236 254 234 238 133 146 169 179 158 180 180 189 179 179 159 149 152 158 156 165 137 124 144 115 105 147 153 138 141 110 119 110 113 130 Ladin
168 193 195 199 224 214 221 622 562 818 760   0 899 839 799 792 656 701 675 731 721 733 737 249 233 244 247 251 238 258 259 254 246 235 244 247 231 198 227 221 207 213 222 222 221 224 214 214 210 214 217 217 209 218 133 146 161 160 145 177 176 175 172 172 151 157 166 167 163 179 146 133 143 119 114 141 151 141 140 114 122 119 122 134 Provencal
163 188 190 194 214 204 210 579 511 803 740 899   0 894 847 825 648 702 667 734 709 730 714 244 233 244 242 246 233 242 244 244 241 230 228 236 236 203 227 221 207 218 233 233 231 235 219 219 221 219 222 228 209 228 133 146 161 161 146 178 177 176 173 167 146 147 155 157 153 168 146 133 143 119 114 141 151 141 139 114 122 120 122 134 French
168 193 195 199 219 209 215 584 525 774 731 839 894   0 781 772 626 673 638 700 685 701 699 249 238 244 242 251 238 247 249 249 246 235 233 241 241 208 222 216 207 217 226 226 226 230 214 223 219 218 217 226 208 228 133 146 166 160 145 177 176 175 172 167 146 147 155 157 153 168 146 133 148 119 114 141 156 135 145 114 122 119 122 134 Walloon
123 147 176 175 201 191 197 513 472 697 653 799 847 781   0 974 577 610 602 643 628 637 627 205 198 209 206 209 194 223 221 215 208 205 203 222 198 184 184 178 158 179 182 182 176 180 171 179 171 175 179 177 166 180 106 121 140 126 120 137 136 141 137 132 125 117 125 122 124 138 120 112  99  83  76 105 111 110 122  83  88  88  88  98 French_Creole_C
124 150 174 173 199 188 195 513 483 689 660 792 825 772 974   0 563 596 582 634 624 639 628 208 196 207 209 218 201 222 219 214 206 208 201 215 201 186 192 186 166 188 190 190 184 188 179 188 179 183 187 186 174 188 107 122 141 128 122 139 138 143 139 134 127 119 127 124 126 141 122 114 101  85  77 106 112 112 123  85  89  89  89  99 French_Creole_D
155 175 194 198 223 212 219 577 528 728 655 656 648 626 577 563   0 796 786 633 640 644 637 241 230 241 241 234 231 224 221 231 234 216 215 223 232 212 213 212 198 216 230 230 227 231 218 206 218 216 219 237 211 236 135 136 153 163 149 175 164 174 160 169 148 144 147 153 151 165 132 119 133  99  99 126 137 126 135 111 120 117 119 131 Sardinian_N
168 188 207 211 236 226 232 602 539 797 687 701 702 673 610 596 796   0 832 667 646 662 651 244 229 239 239 232 229 232 228 239 241 224 223 226 230 209 231 235 211 234 242 242 240 244 230 218 230 228 231 254 230 249 133 145 168 167 147 173 172 177 168 178 168 157 166 172 165 184 135 122 131 108 103 130 145 130 139 120 129 126 128 141 Sardinian_L
154 180 194 197 222 211 218 598 542 731 662 675 667 638 602 582 786 832   0 657 646 646 626 250 235 245 245 233 230 228 224 240 242 221 219 222 236 216 227 226 207 231 239 239 236 240 227 215 227 231 234 246 221 240 134 141 157 157 143 164 163 168 159 168 148 153 156 157 161 174 136 124 132  98  98 125 131 132 129 110 114 122 108 125 Sardinian_C
168 195 187 190 219 209 215 594 531 788 716 731 734 700 643 634 633 667 657   0 874 862 730 253 247 258 256 245 232 241 237 253 250 239 237 230 240 207 226 230 206 228 247 247 240 244 234 218 230 228 231 237 218 232 128 141 166 161 146 179 182 181 173 177 162 157 166 167 163 173 145 132 144 113 108 140 145 141 134 108 117 114 111 128 Spanish
163 183 196 200 219 214 221 629 547 773 731 721 709 685 628 624 640 646 646 874   0 969 708 247 242 253 251 240 232 241 237 258 250 239 237 235 240 212 221 215 196 219 243 243 236 240 230 219 226 224 227 238 219 234 128 146 171 166 151 184 183 187 173 177 157 157 161 167 163 173 146 134 148 119 114 146 151 146 140 109 117 114 117 134 Portuguese_ST
166 193 200 203 223 218 224 608 537 785 722 733 730 701 637 639 644 662 646 862 969   0 706 246 241 251 250 244 230 245 241 251 249 237 234 238 244 216 219 213 204 222 241 241 238 242 227 222 228 227 230 234 216 236 129 143 174 168 153 187 185 189 175 179 159 154 162 163 160 170 147 134 145 120 115 142 147 142 140 115 124 121 118 135 Brazilian
155 176 182 187 207 202 208 577 540 764 696 737 714 699 627 628 637 651 626 730 708 706   0 236 219 236 240 238 225 245 241 246 239 232 241 229 223 195 204 208 199 211 225 225 223 227 215 201 212 216 219 219 206 215 129 138 163 168 153 180 179 179 165 174 153 149 153 154 150 166 147 134 144 120 115 142 147 142 140 110 113 115 118 130 Catalan
179 194 180 185 206 205 206 249 247 265 250 249 244 249 205 208 241 244 250 253 247 246 236   0 858 838 844 837 751 631 624 695 707 633 591 593 578 573 221 224 200 267 238 238 259 258 253 241 247 246 245 255 231 236 111 137 153 151 141 153 147 147 143 163 137 173 188 188 179 210 146 128 149 104 109 140 147 142 134 126 130 131 129 130 German_ST
164 171 156 160 187 187 188 234 229 250 241 233 233 238 198 196 230 229 235 247 242 241 219 858   0 779 800 777 720 571 543 601 624 599 565 576 553 568 190 201 181 230 227 227 238 231 232 219 220 225 223 240 220 220 116 136 148 141 141 160 153 153 144 160 135 163 177 179 169 202 146 134 145 109 114 135 142 142 127 109 114 119 114 113 Penn_Dutch
170 180 160 164 186 185 186 254 249 260 246 244 244 244 209 207 241 239 245 258 253 251 236 838 779   0 965 954 797 648 621 692 663 650 582 582 608 589 200 214 195 246 223 223 244 247 232 215 237 231 224 229 221 221 106 137 152 152 136 154 143 143 144 158 131 173 194 188 184 215 146 139 149 104 103 146 147 142 140 111 109 116 113 114 Dutch_list
169 179 161 165 191 190 191 249 247 259 246 247 242 242 206 209 241 239 245 256 251 250 240 844 800 965   0 970 824 640 618 693 670 646 585 587 605 601 204 218 199 250 233 233 247 251 237 219 241 240 234 239 221 231 100 131 146 136 125 153 142 142 138 147 130 162 177 172 173 204 140 138 154 104 103 145 146 135 133 110 109 116 113 114 Afrikaans
176 181 168 172 188 188 188 232 244 251 234 251 246 251 209 218 234 232 233 245 240 244 238 837 777 954 970   0 844 629 607 680 665 646 583 591 589 589 206 215 196 253 225 225 245 238 234 222 228 232 226 247 228 223 107 138 153 147 137 155 143 143 139 160 132 163 190 179 175 207 141 139 150 105 104 147 153 132 129 116 115 122 120 121 Flemish
165 175 167 171 188 194 194 229 240 245 231 238 233 238 194 201 231 229 230 232 232 230 225 751 720 797 824 844   0 571 564 638 614 606 554 551 561 557 197 202 193 230 210 210 232 231 222 209 226 219 218 235 215 215  99 121 141 136 130 144 132 132 128 154 135 159 174 175 166 198 146 133 124  97 101 114 131 115 114 120 118 125 124 125 Frisian
189 184 165 169 171 174 180 222 236 254 240 258 242 247 223 222 224 232 228 241 241 245 245 631 571 648 640 629 571   0 925 910 830 838 800 811 591 492 213 202 193 246 229 229 233 242 227 236 237 236 245 255 235 236 106 126 158 140 145 163 157 157 142 162 147 168 188 183 168 205 156 144 148  93  98 135 146 130 133 104 103 114 108 114 Swedish_Up
190 184 166 170 177 180 187 218 232 250 236 259 244 249 221 219 221 228 224 237 237 241 241 624 543 621 618 607 564 925   0 904 809 817 784 820 566 453 219 208 194 242 214 214 234 244 223 227 238 227 236 246 226 227 106 127 148 141 141 144 147 142 138 162 141 168 177 178 174 200 152 144 148  93  97 135 146 131 134 120 119 124 118 130 Swedish_VL
191 186 181 186 193 196 202 239 236 259 251 254 244 249 215 214 231 239 240 253 258 251 246 695 601 692 693 680 638 910 904   0 874 842 789 800 589 497 230 218 209 253 230 230 250 259 233 242 254 237 246 257 236 237 100 122 159 141 141 159 152 152 143 163 147 168 184 184 174 206 157 149 148  94  97 135 147 130 133 120 119 125 124 130 Swidish_List
189 183 175 179 191 195 196 237 236 263 249 246 241 246 208 206 234 241 242 250 250 249 239 707 624 663 670 665 614 830 809 874   0 854 779 786 593 518 228 222 203 267 239 239 254 268 247 241 258 251 260 271 240 251 106 126 163 145 140 163 157 157 147 168 152 168 182 183 179 200 151 144 142  98 103 140 151 141 127 121 125 131 130 130 Danish
170 164 151 150 172 176 177 214 222 246 227 235 230 235 205 208 216 224 221 239 239 237 232 633 599 650 646 646 606 838 817 842 854   0 809 821 548 482 205 204 185 238 214 214 236 245 229 228 240 238 242 247 227 228 107 122 153 141 147 149 148 148 138 164 148 164 179 179 170 192 136 129 132  94  98 126 137 126 129 111 110 122 109 120 Riksmal
194 184 180 184 185 189 196 223 229 245 230 244 228 233 203 201 215 223 219 237 237 234 241 591 565 582 585 583 554 800 784 789 779 809   0 922 546 463 208 202 188 237 217 217 234 243 238 232 238 242 246 255 225 232 102 130 157 144 133 147 145 145 141 167 145 182 198 198 194 216 160 148 140 101 106 138 151 145 132 108 107 118 106 117 Icelandic_ST
180 170 173 177 183 188 194 216 218 242 224 247 236 241 222 215 223 226 222 230 235 238 229 593 576 582 587 591 551 811 820 800 786 821 922   0 536 463 196 190 175 219 199 199 221 225 220 229 220 219 228 243 212 219 102 128 144 127 126 140 138 138 134 155 138 180 189 190 192 207 163 157 139  95 100 132 143 138 130 106 105 116 105 115 Faroese
179 183 159 163 169 179 179 227 240 247 239 231 236 241 198 201 232 230 236 240 240 244 223 578 553 608 605 589 561 591 566 589 593 548 546 536   0 658 212 216 197 249 226 226 241 250 219 223 235 239 242 253 228 234 106 131 161 145 135 157 146 146 142 157 141 147 166 162 158 194 130 133 142  93  98 140 147 126 123 113 117 129 122 118 English_ST
156 160 160 164 164 176 176 212 211 215 212 198 203 208 184 186 212 209 216 207 212 216 195 573 568 589 601 589 557 492 453 497 518 482 463 463 658   0 164 168 143 195 185 185 196 195 191 199 189 194 188 203 180 190  85 104 118 114 108 121 109 115 110 121 107 115 134 130 127 153 117 114 108  74  77 101 108 113 114  86  90  95  96  90 Takitaki
191 198 189 188 182 187 188 199 219 234 219 227 227 222 184 192 213 231 227 226 221 219 204 221 190 200 204 206 197 213 219 230 228 205 208 196 212 164   0 899 561 326 376 382 354 370 344 375 361 351 371 386 335 345 157 154 209 177 184 171 173 199 199 195 187 154 167 162 160 191 152 150 132 115 120 161 161 136 110 137 141 142 130 137 Lithuanian_O
189 196 193 186 191 190 192 203 218 242 218 221 221 216 178 186 212 235 226 230 215 213 208 224 201 214 218 215 202 202 208 218 222 204 202 190 216 168 899   0 613 338 394 404 376 395 368 371 373 361 378 390 342 357 156 159 207 181 178 174 168 188 188 194 192 158 182 172 169 205 152 154 133 120 114 161 155 130 109 141 144 147 133 135 Lithuanian_ST
170 176 158 156 155 160 161 179 202 218 209 207 207 207 158 166 198 211 207 206 196 204 199 200 181 195 199 196 193 193 194 209 203 185 188 175 197 143 561 613   0 323 349 354 333 357 323 357 369 332 359 356 306 337 124 118 185 157 152 154 149 152 162 168 160 149 162 152 149 180 131 138 128 103 103 141 141 109 121 124 133 135 122 129 Latvian
180 191 162 171 182 181 182 210 236 240 227 213 218 217 179 188 216 234 231 228 219 222 211 267 230 246 250 253 230 246 242 253 267 238 237 219 249 195 326 338 323   0 679 684 663 694 685 645 610 633 614 691 615 684 147 152 190 183 178 175 186 200 173 199 196 170 188 179 181 203 141 144 133 114 113 151 157 141 128 141 140 164 129 136 Slovenian
173 206 185 188 201 199 200 230 244 250 251 222 233 226 182 190 230 242 239 247 243 241 225 238 227 223 233 225 210 229 214 230 239 214 217 199 226 185 376 394 349 679   0 958 814 800 788 770 720 768 728 795 688 754 163 148 200 185 186 199 199 219 196 202 199 154 178 168 160 177 151 139 135 130 119 168 180 148 133 141 144 153 133 140 Lusatian_L
178 211 185 188 201 199 201 230 244 250 251 222 233 226 182 190 230 242 239 247 243 241 225 238 227 223 233 225 210 229 214 230 239 214 217 199 226 185 382 404 354 684 958   0 830 816 810 780 735 768 743 795 698 749 163 148 206 185 186 199 199 219 197 203 200 154 178 168 160 177 151 139 135 130 119 168 180 148 133 141 150 164 138 151 Lusatian_U
174 212 200 203 211 209 211 223 233 247 240 221 231 226 176 184 227 240 236 240 236 238 223 259 238 244 247 245 232 233 234 250 254 236 234 221 241 196 354 376 333 663 814 830   0 914 883 763 735 766 745 749 689 719 153 144 194 179 176 183 181 203 187 203 194 149 173 164 161 177 159 151 136 126 121 153 169 138 135 130 135 148 130 142 Czech
173 205 194 197 199 198 199 227 237 251 249 224 235 230 180 188 231 244 240 244 240 242 227 258 231 247 251 238 231 242 244 259 268 245 243 225 250 195 370 395 357 694 800 816 914   0 874 813 779 778 741 762 685 732 153 148 198 184 180 181 181 201 186 197 188 155 178 168 171 182 153 151 136 131 115 158 174 147 140 136 141 155 130 142 Slovak
162 189 185 188 197 195 196 214 229 236 234 214 219 214 171 179 218 230 227 234 230 227 215 253 232 232 237 234 222 227 223 233 247 229 238 220 219 191 344 368 323 685 788 810 883 874   0 753 714 737 706 722 672 694 141 138 182 173 169 172 171 192 175 192 181 144 162 158 161 167 153 151 128 126 120 147 163 141 128 136 140 155 131 138 Czech_E
181 214 188 192 194 193 194 201 220 226 223 214 219 223 179 188 206 218 215 218 219 222 201 241 219 215 219 222 209 236 227 242 241 228 232 229 223 199 375 371 357 645 770 780 763 813 753   0 839 802 779 742 660 707 152 137 186 168 173 175 180 195 184 191 181 149 162 158 161 167 156 150 124 120 120 147 168 141 139 131 135 148 124 130 Ukranian
173 200 184 188 196 194 195 212 226 237 234 210 221 219 171 179 218 230 227 230 226 228 212 247 220 237 241 228 226 237 238 254 258 240 238 220 235 189 361 373 369 610 720 735 735 779 714 839   0 751 732 686 646 673 141 138 188 169 180 172 176 192 175 192 182 160 178 174 176 182 153 151 130 126 120 142 158 141 139 126 130 144 124 131 Byelorussian
173 200 178 181 188 188 188 216 237 236 233 214 219 218 175 183 216 228 231 228 224 227 216 246 225 231 240 232 219 236 227 237 251 238 242 219 239 194 351 361 332 633 768 768 766 778 737 802 751   0 734 697 631 680 152 148 196 173 179 181 187 201 185 201 193 155 178 163 171 182 151 150 140 119 124 151 162 135 133 130 129 154 118 130 Polish
192 218 182 186 198 196 197 219 236 239 236 217 222 217 179 187 219 231 234 231 227 230 219 245 223 224 234 226 218 245 236 246 260 242 246 228 242 188 371 378 359 614 728 743 745 741 706 779 732 734   0 720 635 675 151 147 191 183 173 180 180 200 189 200 191 159 177 168 170 181 147 141 133 120 118 156 172 146 133 141 139 164 128 141 Russian
200 207 187 190 202 207 209 222 240 250 254 217 228 226 177 186 237 254 246 237 238 234 219 255 240 229 239 247 235 255 246 257 271 247 255 243 253 203 386 390 356 691 795 795 749 762 722 742 686 697 720   0 835 825 157 153 220 206 189 182 187 207 190 217 205 182 207 201 189 200 152 145 140 130 125 164 180 147 140 157 166 188 157 160 Macedonian
171 182 162 166 177 181 182 202 225 231 234 209 209 208 166 174 211 230 221 218 219 216 206 231 220 221 221 228 215 235 226 236 240 227 225 212 228 180 335 342 306 615 688 698 689 685 672 660 646 631 635 835   0 709 152 148 203 189 183 175 179 199 173 209 197 169 193 189 171 187 131 124 134 119 114 147 163 136 135 139 144 158 139 142 Bulgarian
182 204 179 182 199 198 199 222 236 245 238 218 228 228 180 188 236 249 240 232 234 236 215 236 220 221 231 223 215 236 227 237 251 228 232 219 234 190 345 357 337 684 754 749 719 732 694 707 673 680 675 825 709   0 145 152 191 178 173 175 179 195 173 189 180 170 188 179 182 188 141 134 134 130 120 158 174 147 145 141 145 160 135 142 Serbocroatian
101 109  97  96 101 107 108 134 127 133 133 133 133 133 106 107 135 133 134 128 128 129 129 111 116 106 100 107  99 106 106 100 106 107 102 102 106  85 157 156 124 147 163 163 153 153 141 152 141 152 151 157 152 145   0 188 268 268 267 278 280 305 273 339 292 100  97 100  96 107  91  81 123  96  98 147 161 143 117  91  90  95  90  92 Gypsy_Gk
102  99  87  92  97 102 103 136 144 152 146 146 146 146 121 122 136 145 141 141 146 143 138 137 136 137 131 138 121 126 127 122 126 122 130 128 131 104 154 159 118 152 148 148 144 148 138 137 138 148 147 153 148 152 188   0 242 255 253 237 236 259 279 296 283 108 116 108 108 124 106 102 133  94 100 138 151 148 106  94  93  98  93  93 Singhalese
118 121 129 129 134 138 140 161 171 173 169 161 161 166 140 141 153 168 157 166 171 174 163 153 148 152 146 153 141 158 148 159 163 153 157 144 161 118 209 207 185 190 200 206 194 198 182 186 188 196 191 220 203 191 268 242   0 364 397 394 408 424 410 495 427 119 133 124 125 141 132 132 166 144 160 181 217 218 129 109 113 117 112 112 Kashmiri
118 121 107 112 118 122 123 162 166 167 179 160 161 160 126 128 163 167 157 161 166 168 168 151 141 152 136 147 136 140 141 141 145 141 144 127 145 114 177 181 157 183 185 185 179 184 173 168 169 173 183 206 189 178 268 255 364   0 613 492 469 564 550 552 431 121 129 131 122 142 119 111 148 130 118 166 187 195 145 124 128 124 122 128 Marathi
111 114 101 105 105 110 110 141 144 152 158 145 146 145 120 122 149 147 143 146 151 153 153 141 141 136 125 137 130 145 141 141 140 147 133 126 135 108 184 178 152 178 186 186 176 180 169 173 180 179 173 189 183 173 267 253 397 613   0 543 561 646 535 587 455  94 105 104  94 104 112 104 131 133 131 143 159 192 120 102 106 101 100 105 Gujarati
119 117 115 119 125 130 131 173 172 184 180 177 178 177 137 139 175 173 164 179 184 187 180 153 160 154 153 155 144 163 144 159 163 149 147 140 157 121 171 174 154 175 199 199 183 181 172 175 172 181 180 182 175 175 278 237 394 492 543   0 770 745 551 577 466 112 125 122 118 138 131 124 162 150 150 202 225 254 161 105 109 109 113 114 Panjabi_ST
119 122 114 119 124 130 130 168 168 183 180 176 177 176 136 138 164 172 163 182 183 185 179 147 153 143 142 143 132 157 147 152 157 148 145 138 146 109 173 168 149 186 199 199 181 181 171 180 176 187 180 187 179 179 280 236 408 469 561 770   0 714 521 562 453 118 130 127 124 144 125 118 162 158 155 216 243 251 173 104 108 109 107 113 Lahnda
119 122 124 129 135 140 141 173 172 182 189 175 176 175 141 143 174 177 168 181 187 189 179 147 153 143 142 143 132 157 142 152 157 148 145 138 146 115 199 188 152 200 219 219 203 201 192 195 192 201 200 207 199 195 305 259 424 564 646 745 714   0 641 642 534 117 124 126 118 133 124 117 156 153 154 185 212 240 153 108 112 108 107 112 Hindi
119 122 119 124 130 135 135 168 167 173 179 172 173 172 137 139 160 168 159 173 173 175 165 143 144 144 138 139 128 142 138 143 147 138 141 134 142 110 199 188 162 173 196 197 187 186 175 184 175 185 189 190 173 173 273 279 410 550 535 551 521 641   0 603 511 107 120 117 119 133 115 108 156 133 148 188 204 237 147 109 113 109 108 114 Bengali
128 131 115 119 130 135 135 168 173 178 179 172 167 167 132 134 169 178 168 177 177 179 174 163 160 158 147 160 154 162 162 163 168 164 167 155 157 121 195 194 168 199 202 203 203 197 192 191 192 201 200 217 209 189 339 296 495 552 587 577 562 642 603   0 852 127 135 136 128 138 130 128 144 148 153 174 195 225 142 115 119 119 117 124 Napali_List
115 118 122 122 139 139 140 147 165 168 159 151 146 146 125 127 148 168 148 162 157 159 153 137 135 131 130 132 135 147 141 147 152 148 145 138 141 107 187 192 160 196 199 200 194 188 181 181 182 193 191 205 197 180 292 283 427 431 455 466 453 534 511 852   0 136 138 141 137 141 128 125 112 124 129 143 166 190 113 106 110 109 109 108 Khaskura
133 131 124 128 134 129 130 147 178 168 149 157 147 147 117 119 144 157 153 157 157 154 149 173 163 173 162 163 159 168 168 168 168 164 182 180 147 115 154 158 149 170 154 154 149 155 144 149 160 155 159 182 169 170 100 108 119 121  94 112 118 117 107 127 136   0 923 869 893 673 182 149  99  83  92 126 132 122  99 104 103 109 103 102 Greek_ML
151 150 137 141 147 141 142 150 177 177 152 166 155 155 125 127 147 166 156 166 161 162 153 188 177 194 177 190 174 188 177 184 182 179 198 189 166 134 167 182 162 188 178 178 173 178 162 162 178 178 177 207 193 188  97 116 133 129 105 125 130 124 120 135 138 923   0 918 907 696 190 162 112  90  94 138 138 124 101 117 116 122 115 114 Greek_MD
142 141 133 138 144 138 139 157 183 178 158 167 157 157 122 124 153 172 157 167 167 163 154 188 179 188 172 179 175 183 178 184 183 179 198 190 162 130 162 172 152 179 168 168 164 168 158 158 174 163 168 201 189 179 100 108 124 131 104 122 127 126 117 136 141 869 918   0 868 675 187 154  99  88  93 125 135 122  99 113 117 118 117 118 Greek_Mod
144 142 135 140 146 141 141 154 180 174 156 163 153 153 124 126 151 165 161 163 163 160 150 179 169 184 173 175 166 168 174 174 179 170 194 192 158 127 160 169 149 181 160 160 161 171 161 161 176 171 170 189 171 182  96 108 125 122  94 118 124 118 119 128 137 893 907 868   0 750 178 161 104  89  93 126 127 122  94 111 109 115 109 109 Greek_D
159 159 155 160 166 160 166 168 191 190 165 179 168 168 138 141 165 184 174 173 173 170 166 210 202 215 204 207 198 205 200 206 200 192 216 207 194 153 191 205 180 203 177 177 177 182 167 167 182 182 181 200 187 188 107 124 141 142 104 138 144 133 133 138 141 673 696 675 750   0 188 172 143  94  99 142 148 128 111 130 129 137 128 135 Greek_K
120 122 126 131 126 126 126 136 149 152 137 146 146 146 120 122 132 135 136 145 146 147 147 146 146 146 140 141 146 156 152 157 151 136 160 163 130 117 152 152 131 141 151 151 159 153 153 156 153 151 147 152 131 141  91 106 132 119 112 131 125 124 115 130 128 182 190 187 178 188   0 722 103  90 106 128 144 140 102  80  79  83  84  88 Armenian_Mod
112 109 113 118 112 112 112 118 129 138 124 133 133 133 112 114 119 122 124 132 134 134 134 128 134 139 138 139 133 144 144 149 144 129 148 157 133 114 150 154 138 144 139 139 151 151 151 150 151 150 141 145 124 134  81 102 132 111 104 124 118 117 108 128 125 149 162 154 161 172 722   0  99  87  96 115 126 120  85  82  81  85  86  90 Armenian_List
111 103 101 106 102 112 118 138 139 143 144 143 143 148  99 101 133 131 132 144 148 145 144 149 145 149 154 150 124 148 148 148 142 132 140 139 142 108 132 133 128 133 135 135 136 136 128 124 130 140 133 140 134 134 123 133 166 148 131 162 162 156 156 144 112  99 112  99 104 143 103  99   0 170 185 266 277 229 234  89  88  93  88  99 Ossetic
93 105  93  92  98 103 104  94 103 120 115 119 119 119  83  85  99 108  98 113 119 120 120 104 109 104 104 105  97  93  93  94  98  94 101  95  93  74 115 120 103 114 130 130 126 131 126 120 126 119 120 130 119 130  96  94 144 130 133 150 158 153 133 148 124  83  90  88  89  94  90  87 170   0 749 308 355 286 263  68  73  71  73  77 Afghan
91  93  82  87  92  97  98  98 107 114 105 114 114 114  76  77  99 103  98 108 114 115 115 109 114 103 103 104 101  98  97  97 103  98 106 100  98  77 120 114 103 113 119 119 121 115 120 120 120 124 118 125 114 120  98 100 160 118 131 150 155 154 148 153 129  92  94  93  93  99 106  96 185 749   0 269 289 269 210  71  71  73  71  74 Waziri
104 112 101 105 105 116 116 136 149 141 147 141 141 141 105 106 126 130 125 140 146 142 142 140 135 146 145 147 114 135 135 135 140 126 138 132 140 101 161 161 141 151 168 168 153 158 147 147 142 151 156 164 147 158 147 138 181 166 143 202 216 185 188 174 143 126 138 125 126 142 128 115 266 308 269   0 815 443 402  96  96  99  90  99 Persian_List
120 129 116 121 121 126 127 137 157 157 153 151 151 156 111 112 137 145 131 145 151 147 147 147 142 147 146 153 131 146 146 147 151 137 151 143 147 108 161 155 141 157 180 180 169 174 163 168 158 162 172 180 163 174 161 151 217 187 159 225 243 212 204 195 166 132 138 135 127 148 144 126 277 355 289 815   0 462 442 102 106 107 101 111 Tadzik
100 109  96 101 100 111 111 132 138 136 138 141 141 135 110 112 126 130 132 141 146 142 142 142 142 142 135 132 115 130 131 130 141 126 145 138 126 113 136 130 109 141 148 148 138 147 141 141 141 135 146 147 136 147 143 148 218 195 192 254 251 240 237 225 190 122 124 122 122 128 140 120 229 286 269 443 462   0 352  92  91  94  90  94 Baluchi
82  90 100 105 105 118 118 149 150 140 141 140 139 145 122 123 135 139 129 134 140 140 140 134 127 140 133 129 114 133 134 133 127 129 132 130 123 114 110 109 121 128 133 133 135 140 128 139 139 133 133 140 135 145 117 106 129 145 120 161 173 153 147 142 113  99 101  99  94 111 102  85 234 263 210 402 442 352   0  89  94  93  94 104 Wakhi
95  98  94  94 105 109 109 109 119 115 110 114 114 114  83  85 111 120 110 108 109 115 110 126 109 111 110 116 120 104 120 120 121 111 108 106 113  86 137 141 124 141 141 141 130 136 136 131 126 130 141 157 139 141  91  94 109 124 102 105 104 108 109 115 106 104 117 113 111 130  80  82  89  68  71  96 102  92  89   0 879 840 736 706 Albanian_T
99 107  98  98 108 113 114 113 124 123 119 122 122 122  88  89 120 129 114 117 117 124 113 130 114 109 109 115 118 103 119 119 125 110 107 105 117  90 141 144 133 140 144 150 135 141 140 135 130 129 139 166 144 145  90  93 113 128 106 109 108 112 113 119 110 103 116 117 109 129  79  81  88  73  71  96 106  91  94 879   0 800 734 722 Albanian_Top
104 106  99  98 110 114 115 115 124 120 110 119 120 119  88  89 117 126 122 114 114 121 115 131 119 116 116 122 125 114 124 125 131 122 118 116 129  95 142 147 135 164 153 164 148 155 155 148 144 154 164 188 158 160  95  98 117 124 101 109 109 108 109 119 109 109 122 118 115 137  83  85  93  71  73  99 107  94  93 840 800   0 707 706 Albanian_G
98 101  94  94 104 109 110 113 124 122 113 122 122 122  88  89 119 128 108 111 117 118 118 129 114 113 113 120 124 108 118 124 130 109 106 105 122  96 130 133 122 129 133 138 130 130 131 124 124 118 128 157 139 135  90  93 112 122 100 113 107 107 108 117 109 103 115 117 109 128  84  86  88  73  71  90 101  90  94 736 734 707   0 704 Albanian_K
109 111 104 104 115 120 121 125 129 134 130 134 134 134  98  99 131 141 125 128 134 135 130 130 113 114 114 121 125 114 130 130 130 120 117 115 118  90 137 135 129 136 140 151 142 142 138 130 131 130 141 160 142 142  92  93 112 128 105 114 113 112 114 124 108 102 114 118 109 135  88  90  99  77  74  99 111  94 104 706 722 706 704   0 Albanian_C
```

IE-perc84.txt

```r
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

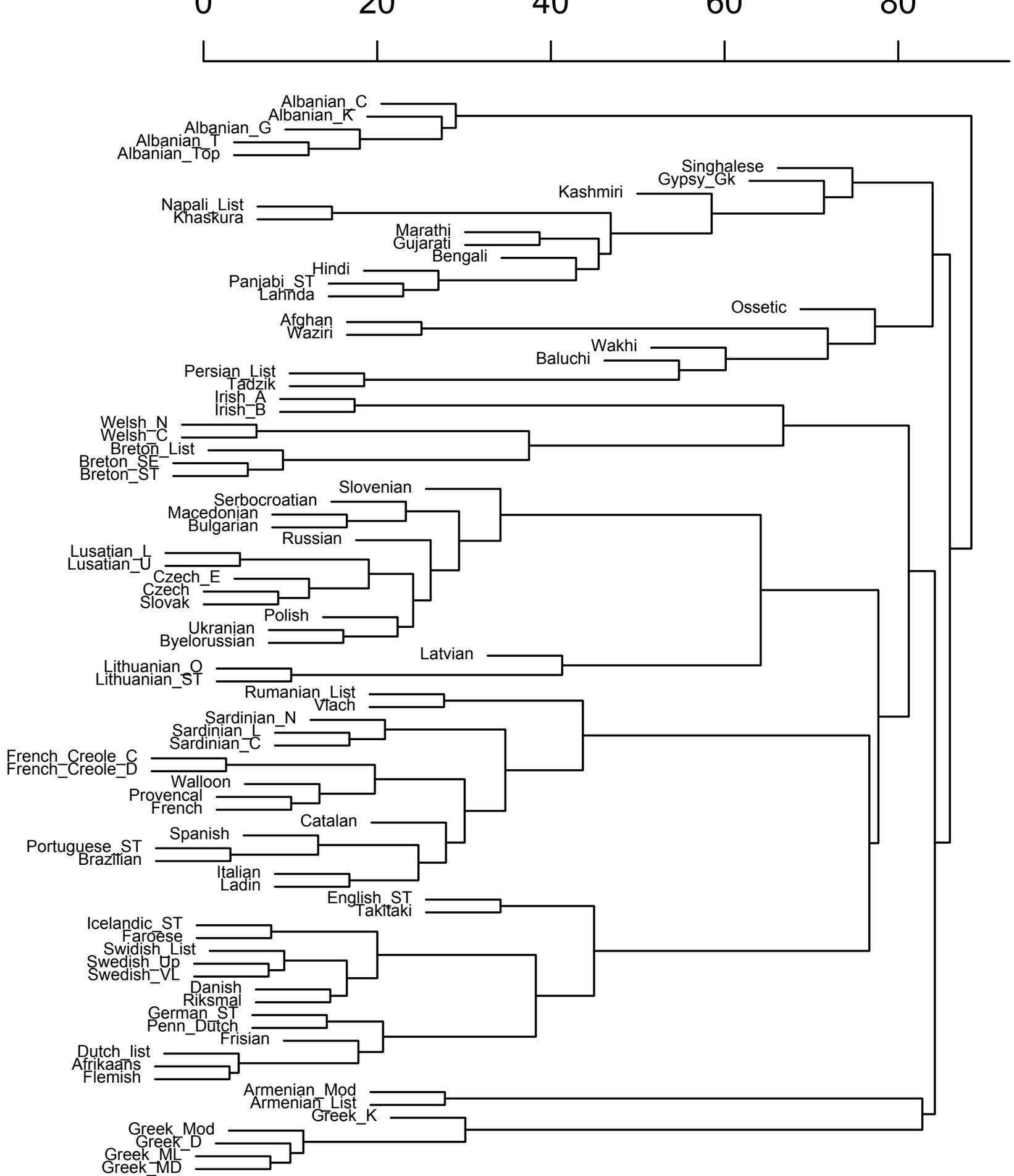
`d`	a dissimilarity structure as produced by `dist`.

```r
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)
```

```r
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)
```

Cluster Dendrogram

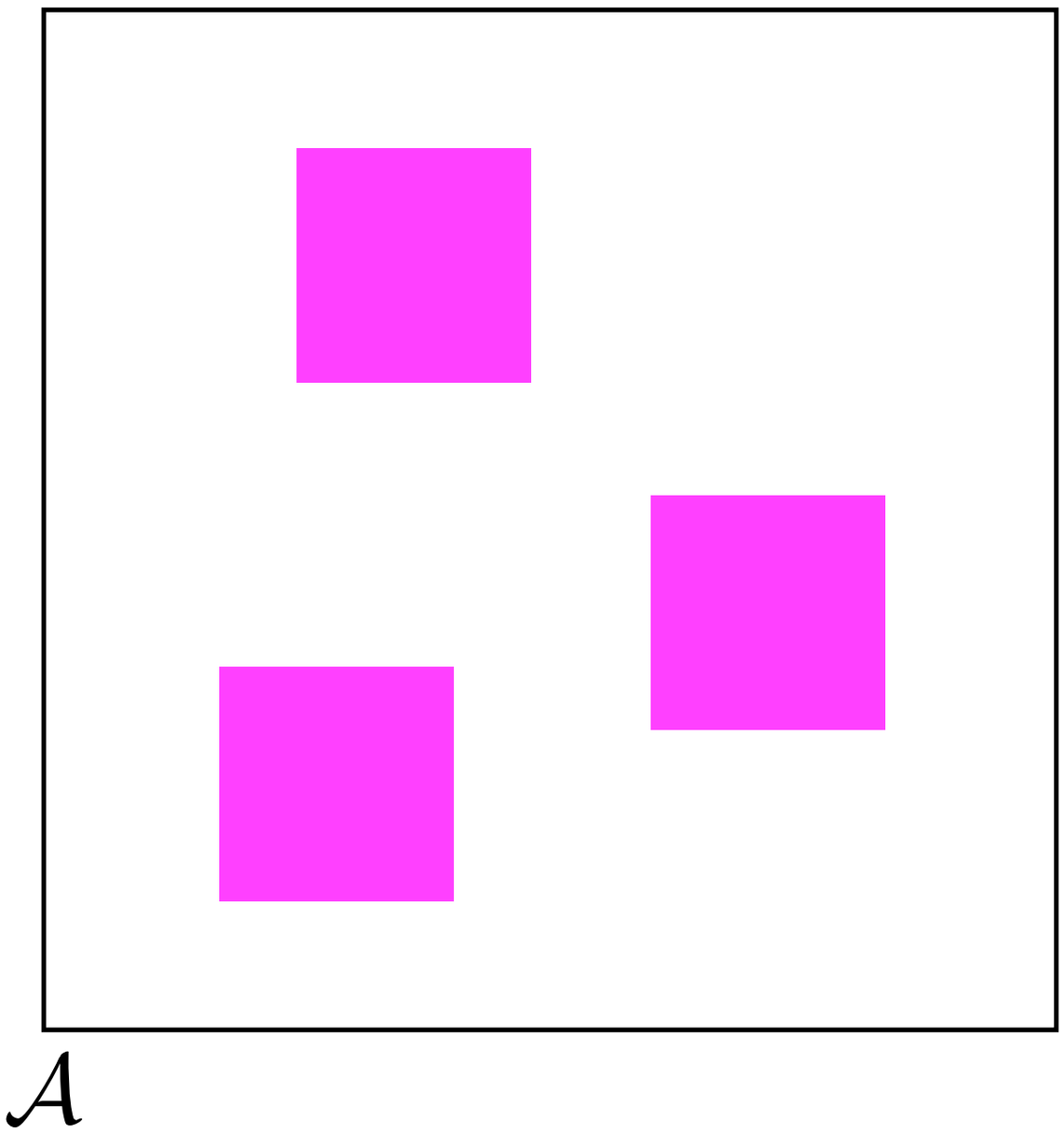
as.dist(d)
hclust (*, "average")

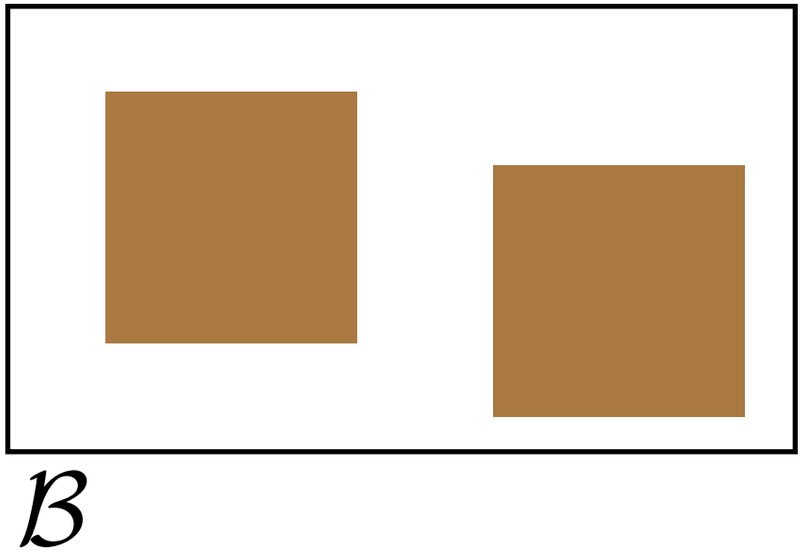


Clustering algorithm

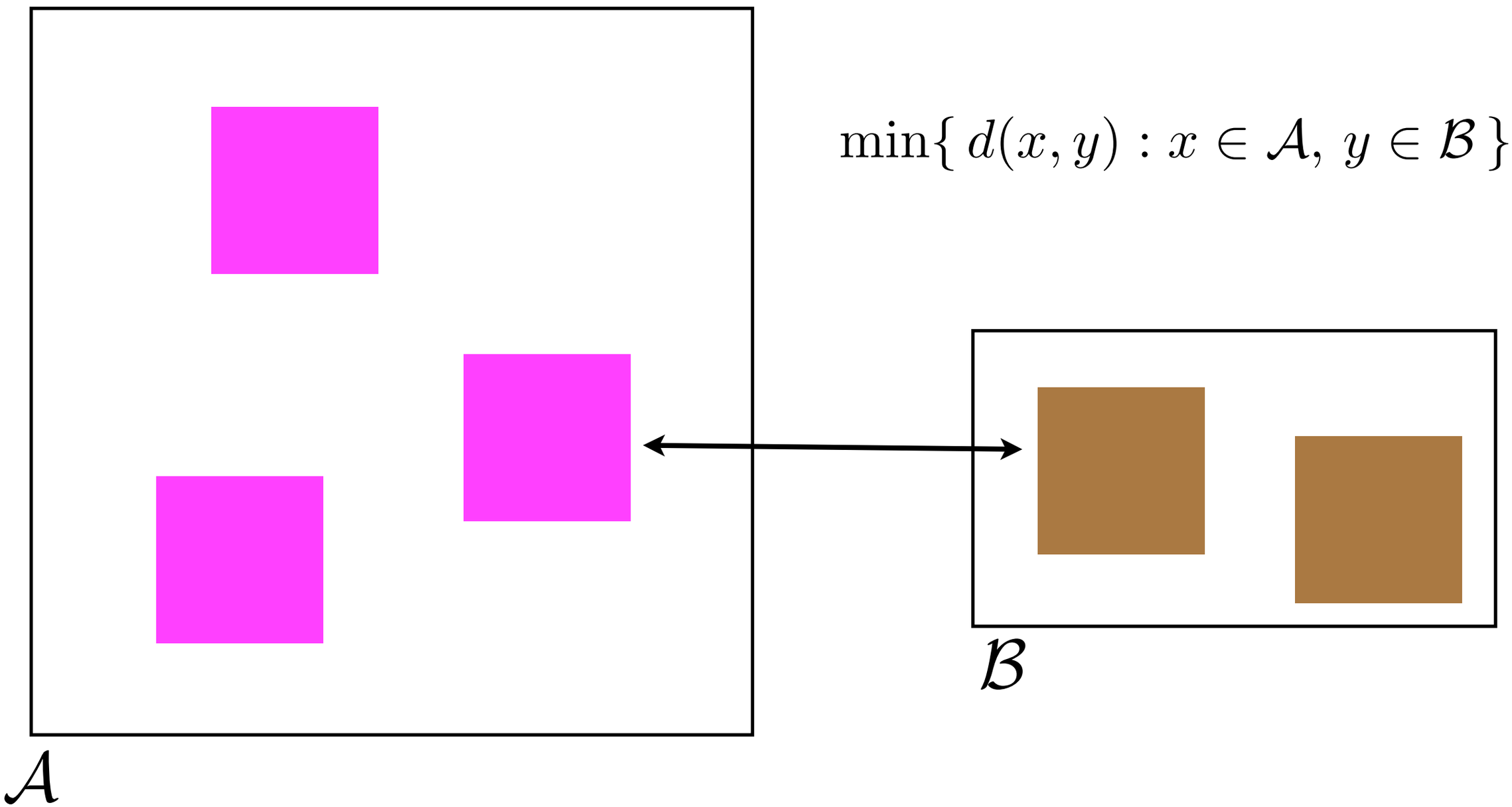
1. Find the two most similar items
2. Link them together into a cluster.
3. Repeat until all items are in a cluster

Basis on which to decide similarity



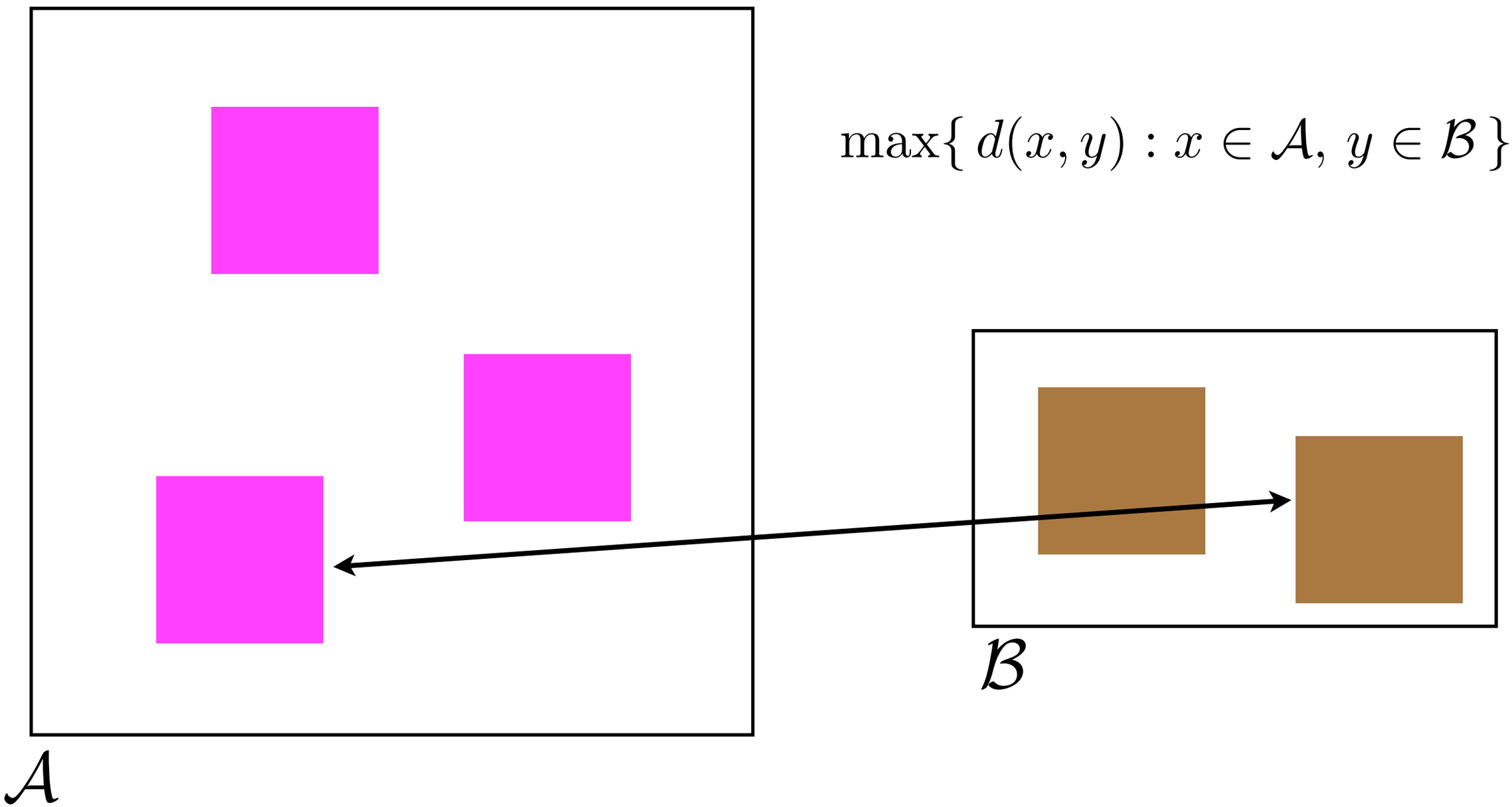


Single linkage



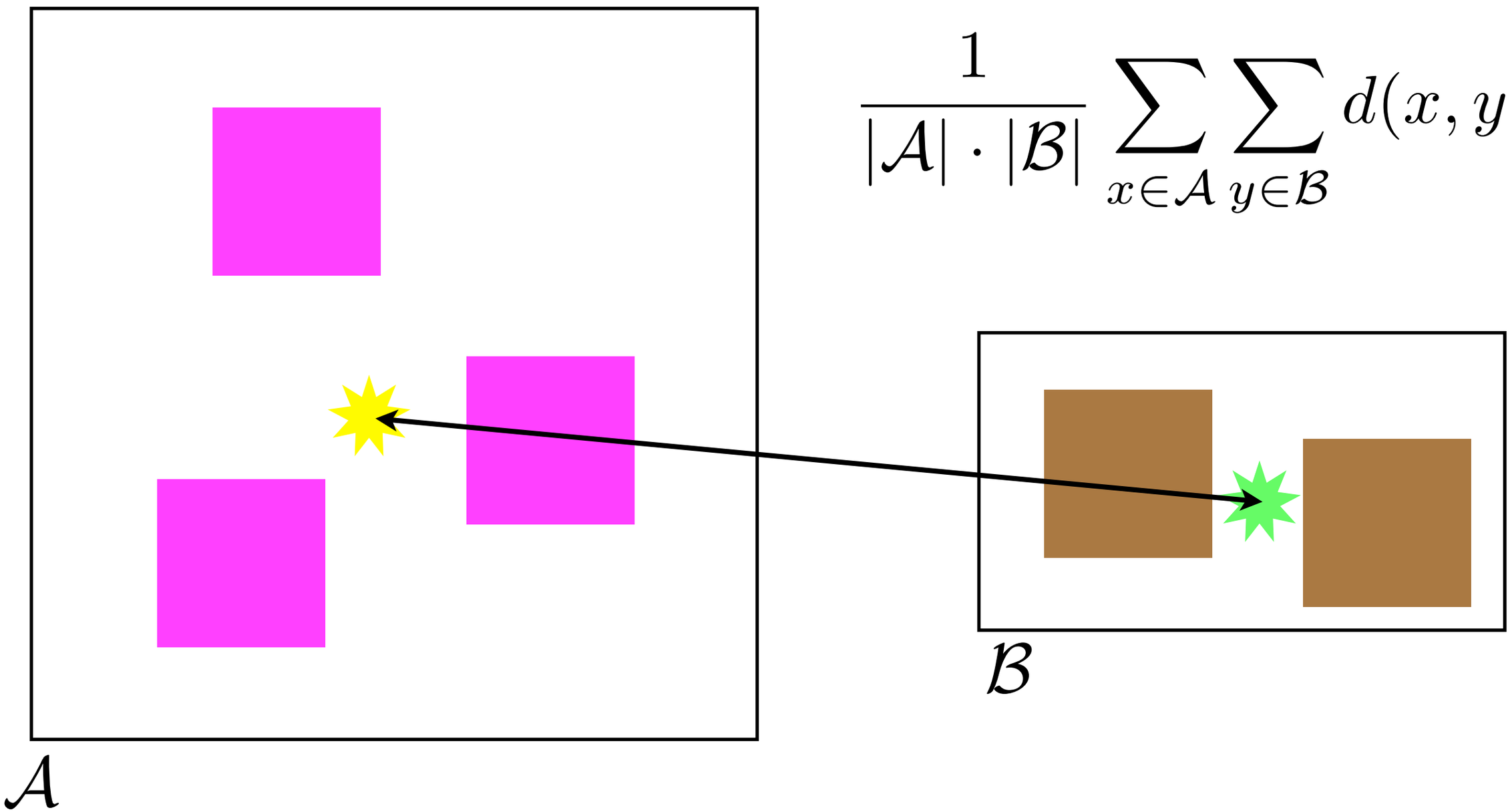
$$\min\{\, d(x,y) : x \in \mathcal{A},\ y \in \mathcal{B} \,\}$$

Complete linkage



$$\max\{\, d(x,y) : x \in \mathcal{A},\ y \in \mathcal{B} \,\}$$

Average linkage



$$\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}} \sum_{y \in \mathcal{B}} d(x, y)$$

Resort to spelling

```
I N T E * N T I O N
| | | | | | | | | |
* E X E C U T I O N
d s s   i s
```

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.

```
i n t e n t i o n
                  ← delete i
n t e n t i o n
                  ← substitute n by e
e t e n t i o n
                  ← substitute t by x
e x e n t i o n
                  ← insert u
e x e n u t i o n
                  ← substitute n by c
e x e c u t i o n
```

Figure 3.24 Transformation list example of Kruskal (1983) from *intention* to *execution*.

Levenshtein distance
aka “dynamic time warping”

$$distance[i,j] = \min \begin{cases} distance[i-1,j] + \text{ins-cost}(target_{i-1}) \\ distance[i-1,j-1] + \text{sub-cost}(source_{j-1}, target_{i-1}) \\ distance[i,j-1] + \text{del-cost}(source_{j-1})) \end{cases}$$

Johnson's price list

$$distance[i,j] = \min \begin{cases} distance[i-1,j] + \text{ins-cost}(target_{i-1}) \\ distance[i-1,j-1] + \text{sub-cost}(source_{j-1}, target_{i-1}) \\ distance[i,j-1] + \text{del-cost}(source_{j-1})) \end{cases}$$

sub-cost:vowel	$1.00
sub-cost:const	$2.00
ins-cost	$1.50
deletion	$1.50

DTW algorithm

```
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., $\text{sub-cost}(x,x) = 0$).

DTW algorithm

```
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]
```

Keith divides this by |n|+|m|

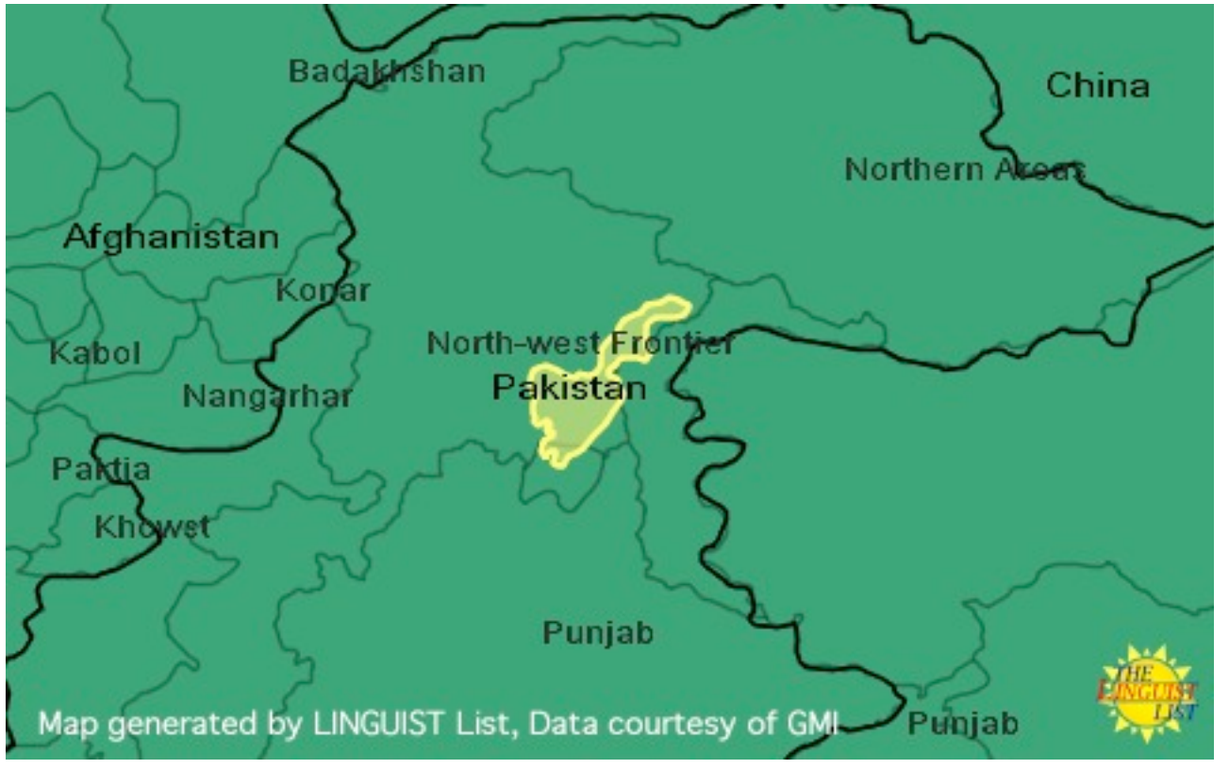
Figure 3.25 The minimum edit distance algorithm, an example of the class of dynamic programming algorithms. The various costs can either be fixed (e.g., $\forall x$, ins-cost$(x) = 1$) or can be specific to the letter (to model the fact that some letters are more likely to be inserted than others). We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

builds *distance* array

n	9	↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙←↓ 12	↓ 11	↓ 10	↓ 9	↙ **8**	
o	8	↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↓ 10	↓ 9	↙ **8**	← 9	
i	7	↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9	**8**	← 9	← 10	
t	6	↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙ **8**	← 9	← 10	←↓ 11	
n	5	↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	←↓ **8**	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙↓ 10	
e	4	↙ 3	← 4	← **5**	← **6**	← 7	←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9	
t	3	↙←↓ 4	↙←↓ **5**	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙ 7	←↓ 8	↙←↓ 9	↓ 8	
n	2	↙←↓ **3**	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↓ 7	↙←↓ 8	↙ 7	
i	**1**	↙←↓ 2	↙←↓ 3	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙ 6	← 7	← 8	
#	**0**	1	2	3	4	5	6	7	8	9	
	#	**e**	**x**	**e**	**c**	**u**	**t**	**i**	**o**	**n**	

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
2;Lahnda;te
3;Lahnda;janwer
4;Lahnda;swa
5;Lahnda;utte
6;Lahnda;picche
7;Lahnda;bura
8;Lahnda;chal
9;Lahnda;kyuke
10;Lahnda;pet
11;Lahnda;wedda
12;Lahnda;perinda
13;Lahnda;ketten
14;Lahnda;kala
15;Lahnda;xun
16;Lahnda;cellen
17;Lahnda;heddi
18;Lahnda;sa ghinnen
19;Lahnda;bha lawen
20;Lahnda;becca
21;Lahnda;beddel
22;Lahnda;thend
23;Lahnda;awen
...
```

```
1;Panjabi ST;seb,sara
2;Panjabi ST;te
3;Panjabi ST;janver
4;Panjabi ST;sva
5;Panjabi ST;te,per
6;Panjabi ST;pitth
7;Panjabi ST;pera
8;Panjabi ST;chell
9;Panjabi ST;kyoke
10;Panjabi ST;pet
11;Panjabi ST;vedda
12;Panjabi ST;penchi
13;Panjabi ST;veddena
14;Panjabi ST;kala
15;Panjabi ST;leu
16;Panjabi ST;vegna
17;Panjabi ST;heddi
18;Panjabi ST;salena
19;Panjabi ST;jelna
20;Panjabi ST;becca
21;Panjabi ST;beddel
22;Panjabi ST;thend
23;Panjabi ST;ona
...
```

...lots of cognates

IE-lists.txt

```
print STDERR "calculating distance for word number (out of 200): ";
foreach $wnum (sort {$a <=> $b} keys %wtable) { # loop over the 200 words
#for ($wnum=4;$wnum<9;$wnum++) { # 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
                $totald{$l1}{$l2}=0;
                $count{$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";
                    }
                }
            }
            $count{$l1}{$l2}++;  # the total is based on "count" words
            # add the min distance between two langs for this word
            $totald{$l1}{$l2}+= $min;
        }
    }
}

foreach $l1 (sort keys %totald) {
    foreach $l2 (sort keys %{$totald{$l1}}) {
        $ave = $totald{$l1}{$l2}/$count{$l1}{$l2};
        print "$l1, $l2, $totald{$l1}{$l2}, $count{$l1}{$l2}, $ave\n";
    }
}
```

get_IE_distance:42

Spelling similarities between Lahnda and Panjabi

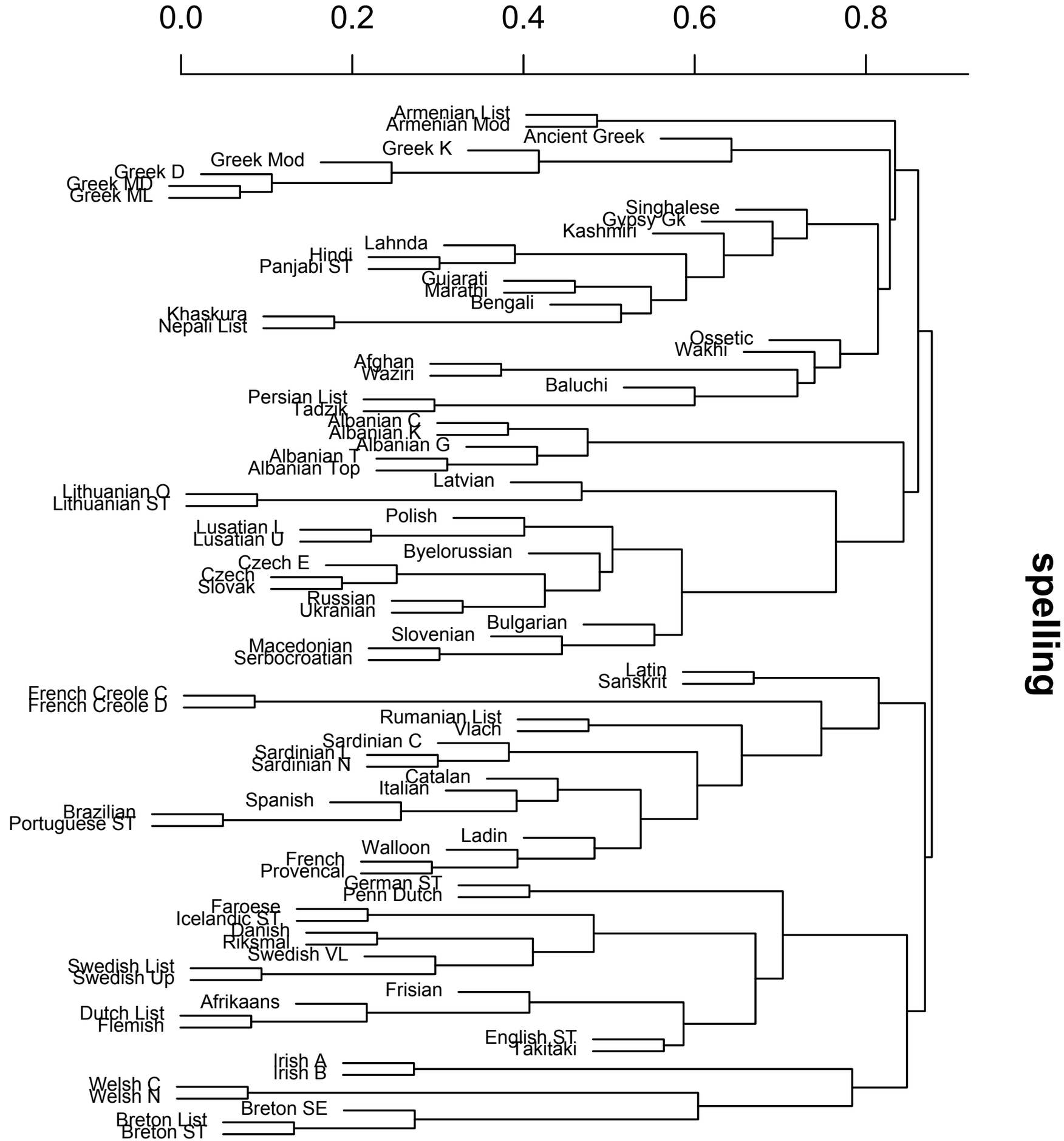
distance between Lahnda and itself is 0

```
0.740,0.773,0.807,0.790,0.775,0.790,0.782,0.789,0.802,0.809,0.698,0.561,0.776,0.772,0.751,0.7
76,0.796,0.812,0.766,0.781,0.789,0.815,0.742,0.820,0.812,0.733,0.784,0.792,0.807,0.782,0.752,
0.793,0.800,0.786,0.801,0.815,0.517,0.691,0.390,0.816,0.856,0.835,0.764,0.594,0.590,0.780,0.0
00,0.806,0.817,0.797,0.799,0.796,0.792,0.767,0.562,0.516,0.754,0.328,0.837,0.696,0.813,0.770,
0.761,0.824,0.791,0.799,0.736,0.781,0.782,0.769,0.797,0.727,0.795,0.791,0.775,0.817,0.818,0.8
33,0.667,0.797,0.791,0.789,0.739,0.779,0.733,0.837,0.829,Lahnda
```

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

IE-distmatrix.txt:47

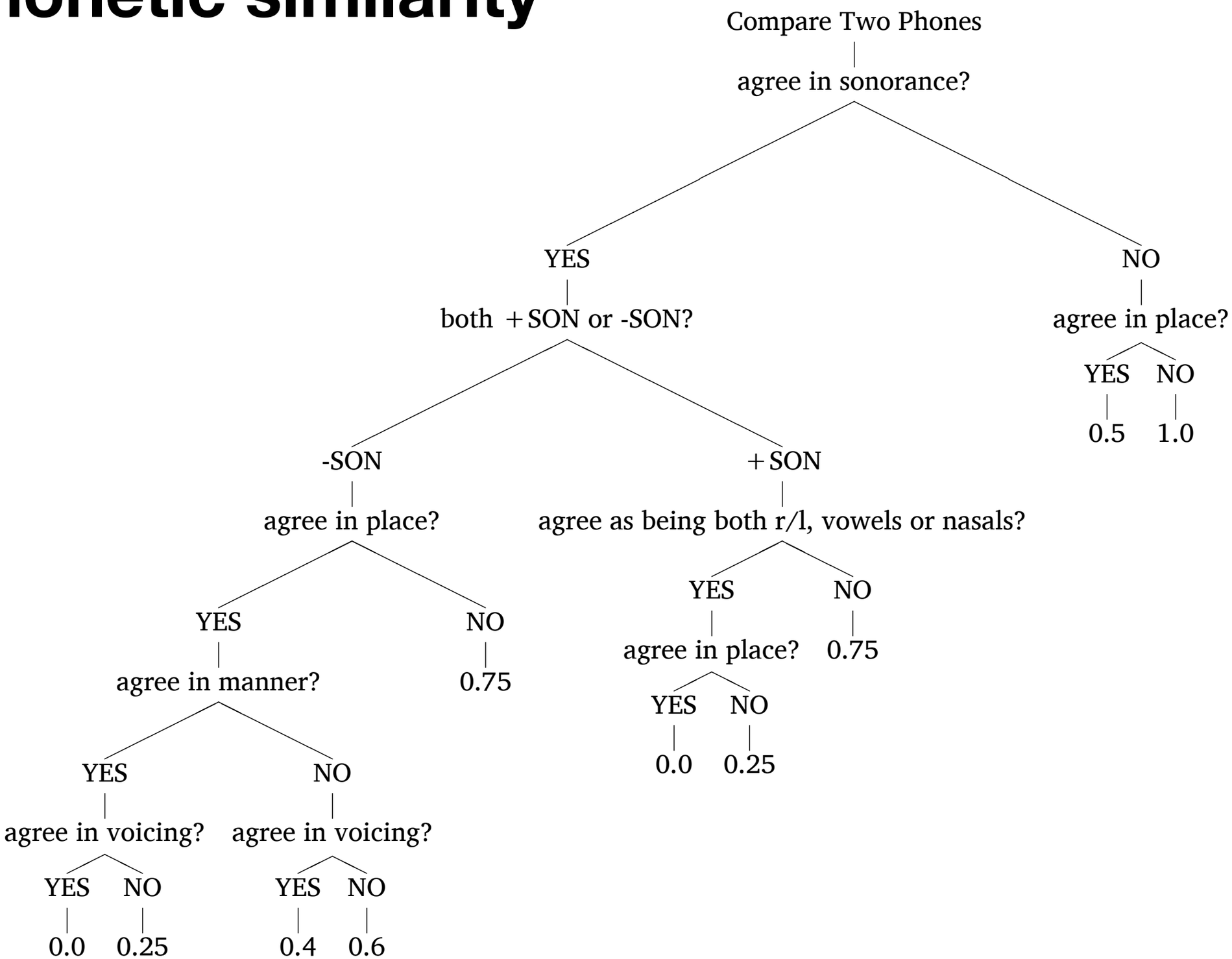
Spelling



hclust(as.dist(d), "complete")

spelling

Phonetic similarity



Sarah Brooks
WashU undergrad honors thesis

Exercise

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