

中間ノードを N, リーフノードを T とする。

ルートノードから順に, 中間ノードに対して以下の処理を再帰的に適用する。

makeGroup(T): 資料群を構成する関数とする。

CT: カレントノードの子供のリーフノードとする。

パス長 n で打ち切る場合, n-パス長の重み付けを行う。

パス長 0: 重み付け α^n

makeGroup(CT)

パス長 1: 重み付け α^{n-1}

for-each child::N

makeGroup(CT | child::T)

パス長 2: 重み付け α^{n-2}

for-each following-sibling::N

makeGroup(CT | child::T)

for-each child::N/child::N

makeGroup(CT | child::T)

パス長 3: 重み付け α^{n-3}

for-each (preceding-sibling::N | following-sibling::N)/child::N

makeGroup(CT | child::T)

for-each child::N/child::N/child::N

makeGroup(CT | child::T)

パス長 4: 重み付け α^{n-4}

for-each parent::N/following-sibling::N/child::N

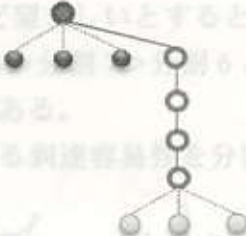
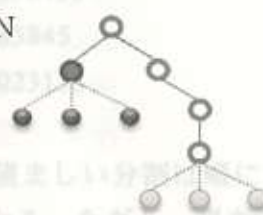
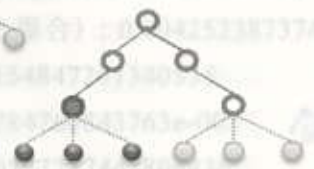
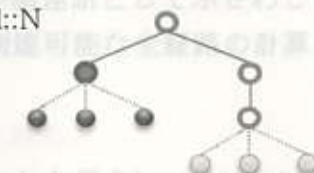
makeGroup(CT | child::T)

for-each (preceding-sibling::N | following-sibling::N)/child::N/child::N

makeGroup(CT | child::T)

for-each child::N/child::N/child::N/child::N

makeGroup(CT | child::T)



part 2 作成
図表作成

木構造を反映した資料群の到達容易性による評価

ある語を入力した場合に、その関連語を関連度の高いものから順に提示することを考え、先頭の語からはじめて、一つ下位の語をみる確率を p とし、 n 番目の語に到達する確率を p^{n-1} 、ある語の関連語を見に行く確率を q (ただし、 $p+q<1$) として、それぞれの資料群設定における到達容易性の計算を行った。 $p=0.89$, $q=p^{20}$ として、関連語は第 60 位まで、別の関連語を辿る回数は 4 回までに制限して、「土器」から「石鐮」に至る到達可能な全経路の計算を行い、全経路の確率の和を求めたところ、

分割 1 (コレクションをそのまま資料群とした場合) : 0.168589930619463

分割 2 (階層構造で資料群を分割した場合) : 0.047174548107095

分割 3 (人手で分割位置を設定した場合) : 0.0744339146642573

分割 4 (階層構造：ルートから深さ方向にゲイン) : 0.0593648767031672

分割 5 (階層構造：パス長 0 のみ) : 0.0439894373190579

分割 6 (階層構造：パス長 1 まで) : 0.0704652440989046

分割 7 (階層構造：パス長 2 まで) : 0.0230753126453718

分割 8 (階層構造：パス長 3 まで) : 0.0133875812892122

分割 9 (階層構造：パス長 4 まで) : 0.00999321824507865

となった。

専門家によって、「石鐮」は「土器」の関連語としてはふさわしくないと判定されているため、「土器」から「石鐮」へ至る到達可能性は低い方が望ましいとすると、望ましい分割は順に、分割 9 > 分割 8 > 分割 7 > 分割 5 > 分割 2 > 分割 4 > 分割 3 > 分割 1 > 分割 6 となる。

次に、「土器」の関連語としては比較的下位で、専門家が関連語としてふさわしいと判定しているものの中から「高杯」を選び、「土器」から「高杯」に至る到達可能な全経路の計算を行い、全経路の確率の和を求めたところ、

分割 1 (コレクションをそのまま資料群とした場合) : 0.00049785187787727

分割 2 (階層構造で資料群を分割した場合) : 0.00142262644849791

分割 3 (人手で分割位置を設定した場合) : 0.00203064928687728

分割 4 (階層構造を反映した計数を行った場合) : 0.00425238737485268

分割 5 (階層構造：パス長 0 のみ) : 0.00154847291340535

分割 6 (階層構造：パス長 1 まで) : 8.73784760843763e-005

分割 7 (階層構造：パス長 2 まで) : 0.000377237442896436

分割 8 (階層構造：パス長 3 まで) : 0.00180140362183845

分割 9 (階層構造：パス長 4 まで) : 0.000785995570023153

となった。こちらは逆に、到達容易性が高いほど望ましいとすると、望ましい分割は順に、分割 4 > 分割 3 > 分割 8 > 分割 5 > 分割 2 > 分割 9 > 分割 1 > 分割 7 > 分割 6 となる。ただし、異なる分割間で、到達容易性の絶対値で比較することには問題もある。

そこで、様々な語から出発して「高杯」に至る到達容易性を分割のそれぞれについて求め、同じ分

割の中でそれらの比較をしてみる。出発語として「石器」、「土器」、「土偶」、「埴輪」、「銅鐸」、「銅剣」、「銅鏡」を設定した場合の計算結果を図2に示す。

	分割1	分割2	分割3	分割4
石器	0.000194056	0.000369437	0.000549275	0.000267481
土器	0.000497852	0.001422626	0.002030649	0.004252387
土偶	0.000345532	0.000186265	0.000621021	0.001287410
埴輪	9.90E-03	0.015456021	0.028347691	0.007692792
銅鐸	0.001193007	0.000708322	0.001994376	0.000154921
銅剣	0.006404267	0.000203992	0.008341554	0.000897488
銅鏡	0.002636213	0.003337144	0.009224251	0.000296631

分割5	分割6	分割7	分割8	分割9
0.000367810	0.000042121	0.000089654	0.000766952	0.000279819
0.001548472	0.000087378	0.000377237	0.001801403	0.000785995
0.000479127	0.000141219	0.000121267	0.000269270	0.000216654
0.024569766	0.002101807	0.000995172	0.000924443	0.000469434
0.001681056	0.000505452	0.000478845	0.000359645	0.000308089
0.007811725	0.003172776	0.002580451	0.002522292	0.002238863
0.009528980	0.000843356	0.001286763	0.001130624	0.001141189

図2 様々な語から「高杯」への到達容易性の分割による違い

それぞれの分割において、到達容易性の高い順に並べてみると、

分割1: 埴輪 > 銅剣 > 銅鏡 > 銅鐸 > 土器 > 土偶 > 石器
 分割2: 埴輪 > 銅鏡 > 土器 > 銅鐸 > 石器 > 銅剣 > 土偶
 分割3: 埴輪 > 銅鏡 > 銅剣 > 土器 > 銅鐸 > 土偶 > 石器
 分割4: 埴輪 > 土器 > 土偶 > 銅剣 > 銅鏡 > 石器 > 銅鐸
 分割5: 埴輪 > 銅鏡 > 銅剣 > 銅鐸 > 土器 > 土偶 > 石器
 分割6: 銅剣 > 埴輪 > 銅鏡 > 銅鐸 > 土偶 > 土器 > 石器
 分割7: 銅剣 > 銅鏡 > 埴輪 > 銅鐸 > 土器 > 土偶 > 石器
 分割8: 銅剣 > 土器 > 銅鏡 > 埴輪 > 石器 > 銅鐸 > 土偶
 分割9: 銅剣 > 銅鏡 > 土器 > 埴輪 > 銅鐸 > 石器 > 土偶

となる。

11922121 10009 11922121
 到達容易性の分割