Intelligibility among Vocabularies of the EU Languages

Jacek Pardyak

5 september 2018

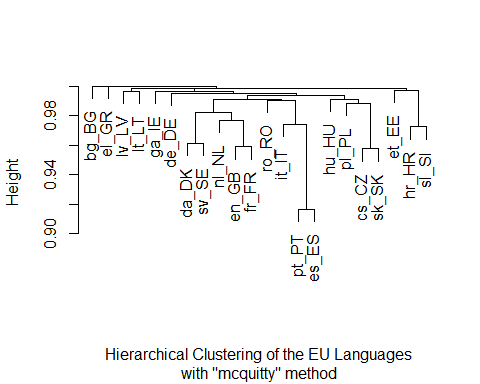
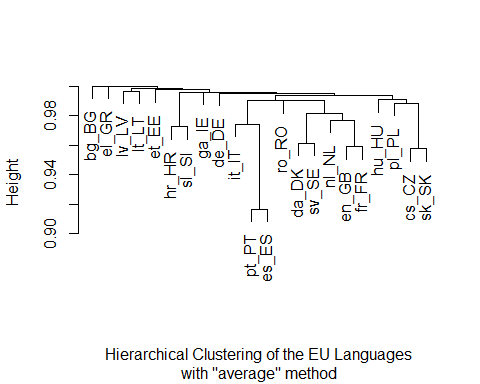
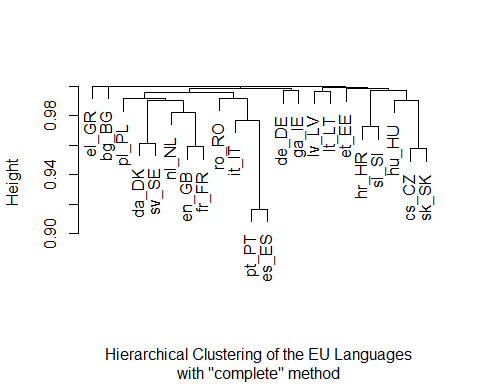
## Introduction

**Clustering European Languages**

In the European Union there are 24 official languages spoken among 28 member states. The EU asserts that it is in favor of linguistic diversity. This principle is a basis of many studies on mutual intelligibility between languages. Relationship between languages in which speakers can readily understand each other without prior familiarity or special effort can be measured with distance function:

For our analysis vocabularies (V) are derived from <http://hunspell.github.io/> spell checker project. For Finnish and Maltese vocabularies are not available. We perform a Hierarchical Cluster Analysis (HCA) using a set of dissimilarities for the 22 vocabularies being clustered. HCA is based on the core idea of objects being more related to nearby objects than to objects farther away. Apart from choice of distance function we also need to decide on the linkage criterion. Linkage Criterion refers to the point from where we merge two clusters. Pictures present results of HCA after applying different linkage criteria.

#mDist2 = 1 - mDist2  
#mDist2 = (1/(mDist2+0.001)) - 1  
#mDist2 = as.dist(mDist2)  
#mDist2



mDF <- as.matrix(mDist)  
pl\_PL <- mDF[, 'pl\_PL']  
pl\_PL[order(pl\_PL, decreasing = FALSE)]

## pl\_PL en\_GB cs\_CZ da\_DK nl\_NL it\_IT sk\_SK   
## 0.0000000 0.9831523 0.9858495 0.9887899 0.9891335 0.9896010 0.9909898   
## sv\_SE fr\_FR hu\_HU sl\_SI es\_ES pt\_PT hr\_HR   
## 0.9912803 0.9916465 0.9923224 0.9923790 0.9943816 0.9944366 0.9946607   
## ro\_RO de\_DE ga\_IE et\_EE lt\_LT lv\_LV bg\_BG   
## 0.9955553 0.9956825 0.9958644 0.9974506 0.9974988 0.9987256 1.0000000   
## el\_GR   
## 1.0000000

nl\_NL <- mDF[, 'nl\_NL']  
nl\_NL[order(nl\_NL, decreasing = FALSE)]

## nl\_NL en\_GB da\_DK fr\_FR sv\_SE pl\_PL it\_IT   
## 0.0000000 0.9713650 0.9807028 0.9822716 0.9866061 0.9891335 0.9909306   
## de\_DE cs\_CZ hu\_HU es\_ES ro\_RO sk\_SK pt\_PT   
## 0.9930614 0.9935128 0.9942315 0.9943269 0.9944476 0.9949912 0.9950583   
## sl\_SI et\_EE ga\_IE hr\_HR lt\_LT lv\_LV bg\_BG   
## 0.9955910 0.9959729 0.9964418 0.9973061 0.9985107 0.9988675 1.0000000   
## el\_GR   
## 1.0000000

en\_GB <- mDF[, 'en\_GB']  
en\_GB[order(en\_GB, decreasing = FALSE)]

## en\_GB fr\_FR nl\_NL sv\_SE da\_DK pt\_PT ro\_RO   
## 0.0000000 0.9588505 0.9713650 0.9726686 0.9729823 0.9792359 0.9797630   
## it\_IT pl\_PL es\_ES ga\_IE hu\_HU de\_DE cs\_CZ   
## 0.9824334 0.9831523 0.9842237 0.9865805 0.9885161 0.9896744 0.9908230   
## sk\_SK sl\_SI et\_EE hr\_HR lt\_LT lv\_LV bg\_BG   
## 0.9932135 0.9937559 0.9970013 0.9971163 0.9975549 0.9986451 1.0000000   
## el\_GR   
## 1.0000000

There are two limitations of this approach:

* we count words in two vocabularies that look the same, but differ significantly in meaning (false friends). For example: ‘hak’ (pl\_PL) - ‘hook’ (en\_GB) but ‘hak’ (nl\_NL) - ‘heel’ (en\_GB)
* we don’t count words in two vocabularies that look similar and mean the same. For example: ‘banan’(pl\_PL) - ‘banaan’(nl\_NL) - ‘banana’(en\_GB)

In the next article I will present hierarchical cluster analysis using ‘partial word match’ instead of ‘exact word match’.

## Definitions

**alphabet** - set of letters

**word** () - sequence of elements from an alphabet

**vocabulary** () - set of words

**words similarity** - two words and are similar if with regard to predetermined distance and threshold .

**common vocabulary** - for two vocabularies and elements of two subsets and meet criterion:

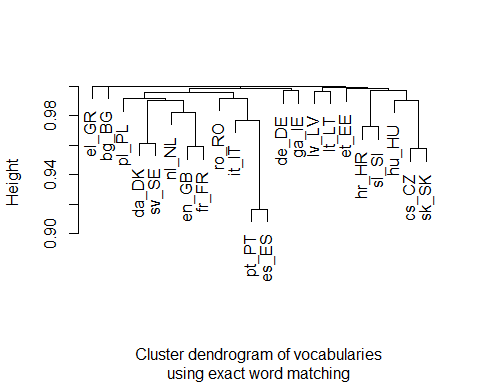
**distance between vocabularies** - ratio of cardinality and cardinality of . More precisely:

In the special case of distance defined as if , and otherwise and threshold , (perfect match) we get

## The data

The EU has 24 official languages used by people within 28 member states. Vocabularies were retrieved from <ftp://ftp.snt.utwente.nl/pub/software/openoffice/contrib/dictionaries/>, part of Hunspell project <http://hunspell.github.io/>. Two of 24 official EU languages, Finnish and Maltese don’t have available vocabularies. We use ISO 639-2 codes of those 22 languages: bg\_BG, hr\_HR, cs\_CZ, da\_DK, nl\_NL, en\_GB, et\_EE, fr\_FR, de\_DE, el\_GR, hu\_HU, ga\_IE, it\_IT, lv\_LV, lt\_LT, pl\_PL, pt\_PT, ro\_RO, sk\_SK, sl\_SI, es\_ES, sv\_SE.

## Clustering vocabularies using exact word matching



mat <- as.matrix(mDist)  
#mat[lower.tri(mat)] <- ''  
knitr::kable(mat[,2:8], caption = "Vocabularies distance using exact word matching")

Vocabularies distance using exact word matching

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | hr\_HR | cs\_CZ | da\_DK | nl\_NL | en\_GB | et\_EE | fr\_FR |
| bg\_BG | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| hr\_HR | 0.0000000 | 0.9950517 | 0.9957717 | 0.9973061 | 0.9971163 | 0.9971855 | 0.9984846 |
| cs\_CZ | 0.9950517 | 0.0000000 | 0.9914238 | 0.9935128 | 0.9908230 | 0.9984447 | 0.9950095 |
| da\_DK | 0.9957717 | 0.9914238 | 0.0000000 | 0.9807028 | 0.9729823 | 0.9959698 | 0.9870562 |
| nl\_NL | 0.9973061 | 0.9935128 | 0.9807028 | 0.0000000 | 0.9713650 | 0.9959729 | 0.9822716 |
| en\_GB | 0.9971163 | 0.9908230 | 0.9729823 | 0.9713650 | 0.0000000 | 0.9970013 | 0.9588505 |
| et\_EE | 0.9971855 | 0.9984447 | 0.9959698 | 0.9959729 | 0.9970013 | 0.0000000 | 0.9976601 |
| fr\_FR | 0.9984846 | 0.9950095 | 0.9870562 | 0.9822716 | 0.9588505 | 0.9976601 | 0.0000000 |
| de\_DE | 0.9996456 | 0.9957151 | 0.9934287 | 0.9930614 | 0.9896744 | 0.9996248 | 0.9971923 |
| el\_GR | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| hu\_HU | 0.9971798 | 0.9904044 | 0.9921764 | 0.9942315 | 0.9885161 | 0.9980162 | 0.9945140 |
| ga\_IE | 0.9993676 | 0.9968137 | 0.9945974 | 0.9964418 | 0.9865805 | 0.9993833 | 0.9963624 |
| it\_IT | 0.9980078 | 0.9961185 | 0.9919548 | 0.9909306 | 0.9824334 | 0.9984985 | 0.9856155 |
| lv\_LV | 0.9983660 | 0.9987971 | 0.9982343 | 0.9988675 | 0.9986451 | 0.9989126 | 0.9990744 |
| lt\_LT | 0.9960027 | 0.9981917 | 0.9977963 | 0.9985107 | 0.9975549 | 0.9987476 | 0.9989904 |
| pl\_PL | 0.9946607 | 0.9858495 | 0.9887899 | 0.9891335 | 0.9831523 | 0.9974506 | 0.9916465 |
| pt\_PT | 0.9983159 | 0.9967711 | 0.9941241 | 0.9950583 | 0.9792359 | 0.9991050 | 0.9914953 |
| ro\_RO | 0.9966747 | 0.9974355 | 0.9939589 | 0.9944476 | 0.9797630 | 0.9986032 | 0.9923275 |
| sk\_SK | 0.9935443 | 0.9575697 | 0.9932409 | 0.9949912 | 0.9932135 | 0.9979164 | 0.9965935 |
| sl\_SI | 0.9725991 | 0.9917212 | 0.9927850 | 0.9955910 | 0.9937559 | 0.9974340 | 0.9973319 |
| es\_ES | 0.9974613 | 0.9975560 | 0.9945903 | 0.9943269 | 0.9842237 | 0.9982971 | 0.9879393 |
| sv\_SE | 0.9951708 | 0.9941117 | 0.9611890 | 0.9866061 | 0.9726686 | 0.9953781 | 0.9907153 |

knitr::kable(mat[,9:15], caption = "Vocabularies distance using exact word matching cntd")

Vocabularies distance using exact word matching cntd

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | de\_DE | el\_GR | hu\_HU | ga\_IE | it\_IT | lv\_LV | lt\_LT |
| bg\_BG | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| hr\_HR | 0.9996456 | 1.0000000 | 0.9971798 | 0.9993676 | 0.9980078 | 0.9983660 | 0.9960027 |
| cs\_CZ | 0.9957151 | 1.0000000 | 0.9904044 | 0.9968137 | 0.9961185 | 0.9987971 | 0.9981917 |
| da\_DK | 0.9934287 | 1.0000000 | 0.9921764 | 0.9945974 | 0.9919548 | 0.9982343 | 0.9977963 |
| nl\_NL | 0.9930614 | 1.0000000 | 0.9942315 | 0.9964418 | 0.9909306 | 0.9988675 | 0.9985107 |
| en\_GB | 0.9896744 | 1.0000000 | 0.9885161 | 0.9865805 | 0.9824334 | 0.9986451 | 0.9975549 |
| et\_EE | 0.9996248 | 1.0000000 | 0.9980162 | 0.9993833 | 0.9984985 | 0.9989126 | 0.9987476 |
| fr\_FR | 0.9971923 | 1.0000000 | 0.9945140 | 0.9963624 | 0.9856155 | 0.9990744 | 0.9989904 |
| de\_DE | 0.0000000 | 1.0000000 | 0.9951991 | 0.9969049 | 0.9977772 | 0.9996399 | 0.9990719 |
| el\_GR | 1.0000000 | 0.0000000 | 0.9999342 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| hu\_HU | 0.9951991 | 0.9999342 | 0.0000000 | 0.9962753 | 0.9946500 | 0.9989956 | 0.9976632 |
| ga\_IE | 0.9969049 | 1.0000000 | 0.9962753 | 0.0000000 | 0.9973169 | 0.9996068 | 0.9991826 |
| it\_IT | 0.9977772 | 1.0000000 | 0.9946500 | 0.9973169 | 0.0000000 | 0.9990711 | 0.9980258 |
| lv\_LV | 0.9996399 | 1.0000000 | 0.9989956 | 0.9996068 | 0.9990711 | 0.0000000 | 0.9966034 |
| lt\_LT | 0.9990719 | 1.0000000 | 0.9976632 | 0.9991826 | 0.9980258 | 0.9966034 | 0.0000000 |
| pl\_PL | 0.9956825 | 1.0000000 | 0.9923224 | 0.9958644 | 0.9896010 | 0.9987256 | 0.9974988 |
| pt\_PT | 0.9960293 | 1.0000000 | 0.9947908 | 0.9955037 | 0.9770610 | 0.9993165 | 0.9978201 |
| ro\_RO | 0.9968151 | 1.0000000 | 0.9962574 | 0.9976936 | 0.9920662 | 0.9994759 | 0.9987267 |
| sk\_SK | 0.9978439 | 1.0000000 | 0.9901538 | 0.9979675 | 0.9960454 | 0.9987204 | 0.9975647 |
| sl\_SI | 0.9976295 | 1.0000000 | 0.9957264 | 0.9972250 | 0.9978785 | 0.9991776 | 0.9991497 |
| es\_ES | 0.9986264 | 1.0000000 | 0.9964403 | 0.9978043 | 0.9713364 | 0.9988941 | 0.9974989 |
| sv\_SE | 0.9920841 | 1.0000000 | 0.9914064 | 0.9963819 | 0.9931007 | 0.9986894 | 0.9972757 |

knitr::kable(mat[,16:22], caption = "Vocabularies distance using exact word matching cntd")

Vocabularies distance using exact word matching cntd

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | pl\_PL | pt\_PT | ro\_RO | sk\_SK | sl\_SI | es\_ES | sv\_SE |
| bg\_BG | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| hr\_HR | 0.9946607 | 0.9983159 | 0.9966747 | 0.9935443 | 0.9725991 | 0.9974613 | 0.9951708 |
| cs\_CZ | 0.9858495 | 0.9967711 | 0.9974355 | 0.9575697 | 0.9917212 | 0.9975560 | 0.9941117 |
| da\_DK | 0.9887899 | 0.9941241 | 0.9939589 | 0.9932409 | 0.9927850 | 0.9945903 | 0.9611890 |
| nl\_NL | 0.9891335 | 0.9950583 | 0.9944476 | 0.9949912 | 0.9955910 | 0.9943269 | 0.9866061 |
| en\_GB | 0.9831523 | 0.9792359 | 0.9797630 | 0.9932135 | 0.9937559 | 0.9842237 | 0.9726686 |
| et\_EE | 0.9974506 | 0.9991050 | 0.9986032 | 0.9979164 | 0.9974340 | 0.9982971 | 0.9953781 |
| fr\_FR | 0.9916465 | 0.9914953 | 0.9923275 | 0.9965935 | 0.9973319 | 0.9879393 | 0.9907153 |
| de\_DE | 0.9956825 | 0.9960293 | 0.9968151 | 0.9978439 | 0.9976295 | 0.9986264 | 0.9920841 |
| el\_GR | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 |
| hu\_HU | 0.9923224 | 0.9947908 | 0.9962574 | 0.9901538 | 0.9957264 | 0.9964403 | 0.9914064 |
| ga\_IE | 0.9958644 | 0.9955037 | 0.9976936 | 0.9979675 | 0.9972250 | 0.9978043 | 0.9963819 |
| it\_IT | 0.9896010 | 0.9770610 | 0.9920662 | 0.9960454 | 0.9978785 | 0.9713364 | 0.9931007 |
| lv\_LV | 0.9987256 | 0.9993165 | 0.9994759 | 0.9987204 | 0.9991776 | 0.9988941 | 0.9986894 |
| lt\_LT | 0.9974988 | 0.9978201 | 0.9987267 | 0.9975647 | 0.9991497 | 0.9974989 | 0.9972757 |
| pl\_PL | 0.0000000 | 0.9944366 | 0.9955553 | 0.9909898 | 0.9923790 | 0.9943816 | 0.9912803 |
| pt\_PT | 0.9944366 | 0.0000000 | 0.9884227 | 0.9967671 | 0.9974409 | 0.9163081 | 0.9929131 |
| ro\_RO | 0.9955553 | 0.9884227 | 0.0000000 | 0.9971572 | 0.9967697 | 0.9902295 | 0.9899504 |
| sk\_SK | 0.9909898 | 0.9967671 | 0.9971572 | 0.0000000 | 0.9930935 | 0.9965448 | 0.9938947 |
| sl\_SI | 0.9923790 | 0.9974409 | 0.9967697 | 0.9930935 | 0.0000000 | 0.9978718 | 0.9935822 |
| es\_ES | 0.9943816 | 0.9163081 | 0.9902295 | 0.9965448 | 0.9978718 | 0.0000000 | 0.9934191 |
| sv\_SE | 0.9912803 | 0.9929131 | 0.9899504 | 0.9938947 | 0.9935822 | 0.9934191 | 0.0000000 |

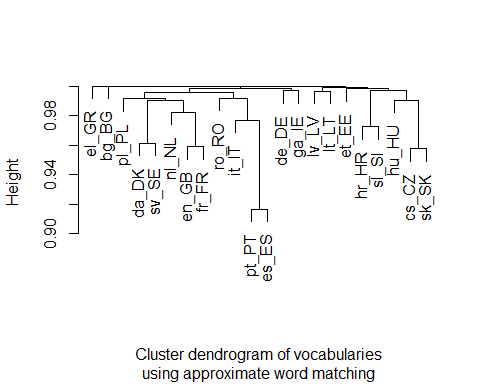
## Clustering vocabularies using approximate word matching

# using Levenshtein distance  
vocab\_1 <- c("leia","uhura", "banan")   
vocab\_2 <- c("ripley","leela","scully","trinity", 'banaan', 'ananas')  
vocab\_3 <- c(NA)  
  
ind <- stringdist::amatch(x = vocab\_1,  
 table = vocab\_2,  
 method = 'lv',  
 maxDist = 3)  
ind

## [1] 2 NA 5

res\_1 = vocab\_1[!is.na(vocab\_2[ind])]  
res\_2 = vocab\_2[ind][!is.na(vocab\_2[ind])]  
  
1 - length(union(res\_1, res\_2))/ length(union(vocab\_1, vocab\_2))

## [1] 0.5555556



## kmed

require(kmed)

## Loading required package: kmed

result <- fastkmed(mDist, ncluster = 5, iterate = 50)