Thematic network analyses of theoretical texts on Bildung and Competence as they relate to science education

Jesper Bruun

2022-06-12

Selection of texts

Four contemporary texts where chosen, two on competences and two on Bildung. Since the analyses are meant to inform Danish science education, we chose two Danish texts as well as two international texts. The two Danish texts have been influential in the Danish academic discussion on Bildung and competence, while the two international texts can be said to represent state-of-the art in the two areas of Bildung and competence.

The Bildung-texts were: Sjöström, J., Frerichs, N., Zuin, V. G., & Eilks, I. (2017). Use of the concept of Bildung in the international science education literature, its potential, and implications for teaching and learning. Studies in Science Education, 53(2), 165-192.

Dolin, J., Jacobsen, L. B., Jensen, S. B., & Johannsen, B. F. (2014). Evaluering af naturvidenskabelig almendannelse i stx-og hf-uddannelserne. Institut for Naturfagenes Didaktik, Københavns Universitet.

The science competence texts were: Ropohl, M., Nielsen, J. A., Olley, C., Rönnebeck, S., & Stables, K. (2018). The concept of competence and its relevance for science, technology and mathematics education. In Transforming assessment (pp. 3-25). Springer, Cham.

Elmose, S. (2015). Naturfaglig Kompetence:–baggrund for begrebet, dets styrker og begrænsninger i naturfagsundervisning.

Selection and translation

We selected the pages in each text, which explicitly describes the concepts of Bildung/competence as they related to science. From Sjostrom et al (2017) we selected pages 167-172, from Dolin et al (2016), we selected pages 12-14 and 16-18. From Ropohl et al (2018), we selected pages 1-10, and finally from Elmose (2015), we selected pages 1-6.

We opted to translate the Danish texts into English using Google Translate as a starting point with subsequent refinements. Before the automatic translation, we translated key phrases such as dannelse and Danish word constructions involving dannelse (such as dannelsesideal) to Bildung and phrases involving Bildung (such as ideal om Bildung). These translations preveserve the meaning in Danish while increasing the quality of the English translation. The preferred way for Google Translate to translate “dannelse” is to “formation”, because the academic translation isn’t very commmon.

Method

See Bruun et al (2019), Bruun et al (2021) and Mariegaard et al (2022) for descriptions of method. Below we describe our workflow for this report.

Load data

English version texts were converted to unformatted .txt files. We load the these files to begin the process:

```
texts <-VCorpus(DirSource("data", encoding = "UTF-8"), readerControl = list(language = "eng"))
```

```
pptexts<-tm_map(texts,tolower)
pptexts<-tm_map(pptexts,removePunctuation)
pptexts<-tm_map(pptexts,stripWhitespace)
wordstoremove<-c(stopwords("english"),"can","also","although","may","maybe","moreover",
                 "mostly","onto","per se", "really","therefore","though","thus",
                 "assistmekudkpracticalexamplesswissexamples","behaving",
                 "bhaskars","biesta","binkley","black","bloemecke","bohlin","bulte","bundsgaard",
                 "burman","bybee","christian","cizek","concept","dale","de","deboer","dolin","duschl",
                 "elmose","elster","erich","etc","fellenz","fischler","føllesdal","gilbert","gitomer",
                 "grugeonallys","gustavsson","hadzigeorgiou","haladyna","hammett","hans","hansgeorg",
                 "harlen","haue","højgaard","horlacher","illeris","james","john","kauertz","kemp",
                 "kivel","kiveloe","klieme","krogh","larsen","leuders","liedman","ling","loevlie",
                 "mascil","masschelein","mead","mertens","nepper","nohl","nordenbo","ørsted","parchmann
                 "paul","poeck","reichenbach","ricken","ricoeur","roberts","roennebeck","roy","rychen",
                 "sadler","said","salganik","schaffar","schecker","schneider","shavelson","shehe","sim"
                 "sjøberg","sodian","sølberg","søren","staeheli","standish","steen","sterling","th","th
                 "theyoeen","thoematiques","thomas","troelsen","uljens","van","vandenabeele","vkevoe","
                 "von","walløe","weinert","weniger","willbergh","wolfgang","wwwestablishfeu","wwwmascil
                 "TRUE","c","compoetences","d","e","eg","fd","handlungsaspekt","ia","ie","knowledge"","

pptexts<-tm_map(pptexts,stripWhitespace)
#Here, we call a script, which will do the substitutions and reductions we want
source("scripts/substitutions.r")


pptexts<-tm_map(pptexts,stripWhitespace)
edgelistTexts<-list()
edgelistTexts[[1]]<-myWordNetwork(pptexts[[1]],j.words=wordstoremove)
edgelistTexts[[2]]<-myWordNetwork(pptexts[[2]],j.words=wordstoremove)
edgelistTexts[[3]]<-myWordNetwork(pptexts[[3]],j.words=wordstoremove)
edgelistTexts[[4]]<-myWordNetwork(pptexts[[4]],j.words=wordstoremove)

#making networks
networkTexts<-list()
for (i in 1:4){
  networkTexts[[i]]<-graph.edgelist(edgelistTexts[[i]],directed=T)
  E(networkTexts[[i]])$weight<-1
  networkTexts[[i]]<-simplify(networkTexts[[i]],remove.multiple=T,remove.loops=T,edge.attr.comb=list(we
  #networkTexts[[1]] is Dolin
  #networkTexts[[2]] is Sjostrom
  #networkTexts[[3]] is Elmose
  #networkTexts[[5]] is Rophol
}

forCollectedEdgelist<-list()
for (i in 1:length(networkTexts)){
  forCollectedEdgelist[[i]]<-get.edgelist(networkTexts[[i]])
}

collectedBildungEdgelist<-rbind(forCollectedEdgelist[[1]],forCollectedEdgelist[[2]])
collectedCompetenceEdgelist<-rbind(forCollectedEdgelist[[3]],forCollectedEdgelist[[4]])
```

```
collectedEdgelist<-do.call(rbind,forCollectedEdgelist)

collectedBildungNetwork<-graph.edgelist(collectedBildungEdgelist,directed=T)
E(collectedBildungNetwork)$weight<-1
collectedBildungNetwork<-simplify(collectedBildungNetwork,remove.multiple=T,remove.loops=T,edge.attr.co

collectedCompetenceNetwork<-graph.edgelist(collectedCompetenceEdgelist,directed=T)
E(collectedCompetenceNetwork)$weight<-1
collectedCompetenceNetwork<-simplify(collectedCompetenceNetwork,remove.multiple=T,remove.loops=T,edge.a

collectedNetwork<-graph.edgelist(collectedEdgelist,directed=T)
E(collectedNetwork)$weight<-1
collectedNetwork<-simplify(collectedNetwork,remove.multiple=T,remove.loops=T,edge.attr.comb=list(weight=

PR<-page.rank(collectedNetwork)
S<-strength(collectedNetwork,mode = "all")
degree<-degree(collectedNetwork)

V(collectedNetwork)$PR<-PR$vector
V(collectedNetwork)$strength<-S
V(collectedNetwork)$degree<-degree
```

Indicate which nodes in single network are part of combined network

Make backbone networks

```
collectedNetworkBB<-backboneNetwork(collectedNetwork,0.01,1) #when replacements have been made, this ma
collectedBildungNetworkBB<-backboneNetwork(collectedBildungNetwork,0.05,1)
collectedCompetenceNetworkBB<-backboneNetwork(collectedCompetenceNetwork,0.05,1)
V(collectedNetworkBB)$bildungWords<-V(collectedNetwork)$bildungWords
V(collectedNetworkBB)$competenceWords<-V(collectedNetwork)$competenceWords
```

#writing graphs

```
write.graph(collectedNetwork, paste0("networks/collectedNetwork", format(Sys.time(), "%d-%b-%Y-%H-%M"),

write.graph(collectedNetwork, paste0("networks/collectedNetwork", format(Sys.time(), "%d-%b-%Y-%H-%M"),

write.graph(collectedBildungNetworkBB, paste0("networks/collectedBildungNetworkBB", format(Sys.time(),

write.graph(collectedBildungNetworkBB, paste0("networks/collectedBildungNetworkBB", format(Sys.time(),

write.graph(collectedCompetenceNetworkBB, paste0("networks/collectedCompetenceNetworkBB", format(Sys.ti

write.graph(collectedCompetenceNetworkBB, paste0("networks/collectedCompetenceNetworkBB", format(Sys.ti

write.graph(collectedNetworkBB, paste0("networks/collectedNetworkBB", format(Sys.time(), "%d-%b-%Y-%H-%

write.graph(collectedNetworkBB, paste0("networks/collectedNetworkBB", format(Sys.time(), "%d-%b-%Y-%H-%
```

Finding module solutions

```
#find a membership solution
memFG<-fastgreedy.community(as.undirected(collectedNetwork)) #fast and greedy may not be optimal due to
#memIM<-infomap.community(collectedNetwork) #igraphs version of infomap may not be optimal
memFGBB<-fastgreedy.community(as.undirected(collectedNetworkBB)) #fast and greedy may not be optimal du
```

The implementations of different community detection in igraph rely on undirected networks. Thus, we will loose information by using these. Instead, we use infomap in a terminal, and make 1000 infomap solutions to test reliability. We will then use the most common solution.

The code used in terminal is: for i in {1..1000};do ./Infomap –clu –ftree -d networks/collectedNetworkBB02-Mar-2022-13-55.net –out-name sol$i infomapSolutions;done;

Making a map of bildung and competencies

```
#consider running infomap as api or just "by hand"
collectedMap<-makemap(memFG$membership,collectedNetwork)
collectedMapBB<-backboneNetwork(collectedMap,0.001,1)
V(collectedMapBB)$n_words<-V(collectedMap)$n_words
V(collectedMapBB)$internallinks<-V(collectedMap)$internallinks

write.graph(collectedMapBB, paste0("networks/collectedMapBB", format(Sys.time(), "%d-%b-%Y-%H-%M"), ".n

write.graph(collectedMapBB, paste0("networks/collectedMapBB", format(Sys.time(), "%d-%b-%Y-%H-%M"), ".g

#Create module networks
#modulesCollected<-graphModules(collectedNetwork,memIM$V2)
#for(i in 1:length(unique(memIM$V2))){
#write.graph(modulesCollected[[i]], paste0("networks/moduleNetworks/M",i,"_", format(Sys.time(), "%d-%b
#  }


#Find out how much each separate text network loads on each module
#networkTexts2<-list(collectedBildungNetwork,collectedCompetenceNetwork)
#loadsTextsOnCollectedNetwork<-list()
#for(i in 1:2){
#loadsTextsOnCollectedNetwork[[i]]<-loads(i,networkTexts2,memFG$membership,modulesCollected)
#}
#wordLoadMatrix<-matrix(0,nrow=2,ncol=length(unique(memFG$membership)))
#linkLoadMatrix<-matrix(0,nrow=2,ncol=length(unique(memFG$membership)))

#for(j in 1:length(networkTexts2)){
 # wordLoadMatrix[j,]<-loadsTextsOnCollectedNetwork[[j]][,1]
 # linkLoadMatrix[j,]<-loadsTextsOnCollectedNetwork[[j]][,2]
#}
```

Finding out how much each text loads on modules in map

Heatmaps

```
#Heatmap
#colnames(wordLoadMatrix) <- paste("M", 1:length(unique(memFG$membership)), sep="")
#rownames(wordLoadMatrix) <- c("Bildung","Competence")
#m<-round(wordLoadMatrix,digits = 1)
```

```
#colnames(linkLoadMatrix) <- paste("M", 1:length(unique(memFG$membership)), sep="")
#rownames(linkLoadMatrix) <- c("Bildung","Competence")
#q<-round(linkLoadMatrix,digits = 1)

#image(1:ncol(m), 1:nrow(m), t(m), col = rev(heat.colors(60)), axes = FALSE)
#axis(1, 1:ncol(m), colnames(m))
#axis(2, 1:nrow(m), rownames(m))
#for (x in 1:ncol(m))
 # for (y in 1:nrow(m))
   # text(x, y, m[y,x])

#writing word lists
```

```
write.csv(V(collectedNetwork)$name, paste0("networks/collectedWordlist", format(Sys.time(), "%d-%b-%Y %

write.xlsx(V(collectedNetwork)$name, paste0("networks/collectedWordlist", format(Sys.time(), "%d-%b-%Y
```

Todo

- calc NMI and make co-module matrix
- make module networks
- make name modules
- heatmaps