

**INSTITUT D’ÉTUDES** **POLITIQUES DE PARIS**

**PARIS SCHOOL OF INTERNATIONAL AFFAIRS**

MASTER IN INTERNATIONAL PUBLIC MANAGEMENT

**USING AUGMENTED INTELLIGENCE IN ACCELERATING THE ERADICATION OF MODERN SLAVERY**

APPLIED MACHINE LEARNING IN ANALYSING AND BENCHMARKING THE MODERN SLAVERY BUSINESSES’ REPORTS

**THE CODE BOOK**

ADRIANA-EUFROSINA BORA

*Thesis directed by Jean-Philippe COINTET, Associate Professor at Paris School of International Affairs (PSIA) and adjunct research scholar at INCITE, Columbia University*

MARCH 2019

*A.E. BORA*

*adrianaeufrosina.bora@sciencespo.fr*

The copyright of this Master's thesis remains the property of its author. No part of the content may be reproduced, published, distributed, copied or stored for public or private use without written permission of the author. All authorisation requests should be sent to vanessa.scherrer@sciencespo.fr

**STM: 6 MONTHS APARAT DATA ANALYSIS**

FEBRUARY DATA:

**###DOWNLOAD THAT WORKS MAINLY FOR THE ULR BUT DOWNLOADS THE PDFS EMPTY :**i <- 0  
for (url in modernslaveryregistry\_2019\_02\_13$URL)   
{i <- i+1  
  
if (grepl('.pdf', url)){  
 filename = paste(toString(i), 'pdf', sep=".")  
}  
else  
{  
 filename = paste(toString(i), 'html', sep=".")  
}  
filename=paste("C:/Users/Utilisateur/Desktop/1",filename,sep='/')  
print (filename)  
try(download.file(url=url,destfile=filename, mode = "wb",  
 cacheOK = TRUE))  
}  
  
**###Downloading PDF THAT WORKS**i <- 0  
for (url in modernslaveryregistry\_2019\_02\_13$URL)   
{i <- i+1  
  
if (grepl('.pdf', url)){  
 filename = paste(toString(i), 'pdf', sep=".")  
}  
  
filename=paste("C:/Users/Utilisateur/Desktop/2",filename,sep='/')  
print (filename)  
try(download.file(url=url,destfile=filename, mode = "wb",  
 cacheOK = TRUE))  
}  
  
  
**###PDF TO TEXT**pdflist<-list.files(pattern="\\.(pdf)$")  
length(pdflist)  
  
folder<-file.path("C:/Users/Utilisateur/Desktop/2")  
folder  
length<-length(dir(folder))  
length  
dirpdf<-dir(folder)  
dirpdf[1]  
pdftotxt<-"C:/Users/Utilisateur/Desktop/xpdf-tools-win-4.01/xpdf-tools-win-4.01/bin32/pdftotext.exe"  
  
for(i in 1:length(dir(folder)))  
{  
 pdf<-file.path("C:/Users/Utilisateur/Desktop/2", dirpdf[i])  
 system(paste("\"", pdftotxt, "\" \"", pdf, "\"", sep = ""), wait = F)  
}  
  
  
  
**###HTML TO TEXT**  
  
DATA\_DIR <- system.file("extdata/", package = "readtext")  
  
  
folder1<-file.path("C:/Users/Utilisateur/Desktop/1")  
folder1  
length<-length(dir(folder1))  
length  
dirhtml<-dir(folder1)  
dirhtml[1]  
  
  
for(i in 1:length(dir(folder1)))  
{   
   
 html<- file.path("C:/Users/Utilisateur/Desktop/1", dirhtml[i])  
 txt<- htmlToText(html)  
 write.csv(txt,file=paste0("C:/Users/Utilisateur/Desktop/",sub(".html","",dirhtml[i]),".txt"))  
   
 }  
  
 **FUNCTION HTMLTOTEXT**   
htmlToText <- function(input, ...) {  
 ###---PACKAGES ---###  
 require(RCurl)  
 require(XML)  
   
   
 ###--- LOCAL FUNCTIONS ---###  
 # Determine how to grab html for a single input element  
 evaluate\_input <- function(input) {   
 # if input is a .html file  
 if(file.exists(input)) {  
 char.vec <- readLines(input, warn = FALSE)  
 return(paste(char.vec, collapse = ""))  
 }  
   
 # if input is html text  
 if(grepl("</html>", input, fixed = TRUE)) return(input)  
   
 # if input is a URL, probably should use a regex here instead?  
 if(!grepl(" ", input)) {  
 # downolad SSL certificate in case of https problem  
 if(!file.exists("cacert.perm")) download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.perm")  
 return(getURL(input, followlocation = TRUE, cainfo = "cacert.perm"))  
 }  
   
 # return NULL if none of the conditions above apply  
 return(NULL)  
 }  
   
 # convert HTML to plain text  
 convert\_html\_to\_text <- function(html) {  
 doc <- htmlParse(html, asText = TRUE)  
 text <- xpathSApply(doc, "//text()[not(ancestor::script)][not(ancestor::style)][not(ancestor::noscript)][not(ancestor::form)]", xmlValue)  
 return(text)  
 }  
   
 # format text vector into one character string  
 collapse\_text <- function(txt) {  
 return(paste(txt, collapse = " "))  
 }  
   
 ###--- MAIN ---###  
 # STEP 1: Evaluate input  
 html.list <- lapply(input, evaluate\_input)  
   
 # STEP 2: Extract text from HTML  
 text.list <- lapply(html.list, convert\_html\_to\_text)  
   
 # STEP 3: Return text  
 text.vector <- sapply(text.list, collapse\_text)  
 return(text.vector)  
}

**###next step: I used cortext to create a dataset "text " with all the documents .txt   
###next I merged initial data set with the txt dataset :**

total <- merge(modernslaveryregistry\_2019\_02\_13,text,by="filename")  
summary(total)  
library(tidyr)  
total %>% drop\_na(text)  
Data has 6795 obs, compared with the initail data set with 9086   
  
  
 ###Packages :   
 install.packages("dplyr")  
install.packages("tidyr")  
install.packages("stringr")  
install.packages("tidytext")  
install.packages("stm")  
install.packages("igraph")  
install.packages("stmCorrViz")  
  
library(dplyr)  
library(tidyr)  
library(stringr)  
library(tidytext)  
library(stm)  
library(igraph) # Package for network analysis and visualisation  
library(stmCorrViz) # Package for hierarchical correlation view of STMs  
  
**###Reading and processing text data/ Prepare: Associating text with metadata**  
processed <- textProcessor(total$text, metadata = total)  
 out <- prepDocuments(processed$documents, processed$vocab, processed$meta)  
 docs <- out$documents  
 vocab <- out$vocab  
 meta <- out$meta  
  
   
 **###selectmodel**  
   
Select<-selectModel(out$documents,out$vocab,K=20,prevalence = ~`UK Modern Slavery Act` + `California Transparency in Supply Chains Act`+ Industry + `Period Covered`, max.em.its = 75, data = out$meta,runs = 20,seed=845819)  
 plotModels(Select,pch=c(1,2,3,4), legend.position = "bottomright")  
   
 selected.1 <- Select$runout[[1]]  
 selected.2 <- Select$runout[[2]]  
   
 selected.3<- Select$runout[[3]]  
 selected.4<- Select$runout[[4]]  
   
 topicQuality(selected.1, documents = docs, main="Model1")  
 topicQuality(selected.2, documents = docs,main="Model2")  
 topicQuality(selected.3, documents = docs, main="Model3")  
 topicQuality(selected.4, documents = docs, main="Model4")  
   
   
   
**### ManyTopics**  
 install.packages("devtools")  
   
 set.seed(02139)  
   
 storage<-manyTopics(docs,vocab,K=3:20, prevalence=~`UK Modern Slavery Act` + `California Transparency in Supply Chains Act`+ Industry + `Period Covered`,data=meta, runs=10)  
   
 text <- out$documents[-c(as.integer(processed$docs.removed))][-c(as.integer(out$docs.removed))]  
   
 print\_models(storage, text, file = "manytopics\_feb.pdf",  
 title = "manytopics\_feb")  
   
 t3<-storage$out[[1]]  
 t4<-storage$out[[2]]  
 ….

t20<-storage$out[[18]]  
   
 plot(t3, type = "summary", main= "t3")  
 plot(t4, type="summary", main= "t4")  
 ….  
 plot(t20, type="summary", main= "t20")  
   
 topicQuality(t3, documents = docs, main="t3")  
 topicQuality(t4, documents = docs, main="t4")  
…  
 topicQuality(t20, documents = docs, main="t20")  
   
 devtools::install\_github('cschwem2er/stminsights')  
 install.packages('stminsights')  
   
 library(stminsights)  
 run\_stminsights()  
   
  
 prep4<- estimateEffect(1:20 ~`UK Modern Slavery Act` + `California Transparency in Supply Chains Act`+ Industry + s(`Period Covered`), selected.4, meta = out$meta, uncertainty ="Global")

summary(prep4)

prep4\_industry<- estimateEffect(1:20 ~ Industry, selected.4, meta = out$meta, uncertainty ="Global")

plot(prep4, "Industry", model = NULL, topics = 1,

method = c("pointestimate"), width = 50, main = "Topic 17", text.cex = 0.8, maxwidth=400)

plot(prep4, covariate = "Period Covered", topics = c(1,8,10,12,15,19), model = selected.4, method="pointestimate", order(covariate, decreasing = F)

**My metric creation:**

library(tokenizers)

**##WFF+MSA Impact on Company Behaviour"**

bigvector=c()  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])  
{ifelse(  
grepl("Updating the company’s code of conduct",sent, ignore.case = TRUE)|   
grepl("Updating the supplier code",sent, ignore.case = TRUE)|   
grepl("new modern slavery company policy ",sent, ignore.case = TRUE)|   
grepl("Updating supplier contracts",sent, ignore.case = TRUE)|   
grepl("new training programme",sent, ignore.case = TRUE)|   
grepl(" new key performance indicators",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA Impact on Company Behaviour'=bigvector

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])   
{ifelse(  
grepl("Updating the company’s code of conduct",sent, ignore.case = TRUE)|   
grepl("Updating the supplier code",sent, ignore.case = TRUE)|   
grepl("new modern slavery company policy ",sent, ignore.case = TRUE)|   
grepl("Updating supplier contracts",sent, ignore.case = TRUE)|   
grepl("new training programme",sent, ignore.case = TRUE)|   
grepl(" new key performance indicators",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA Impact on Company Behaviour1'=bigvector  
Only\_clean\_reports$'WFF+MSA Impact on Company Behaviour1' <- as.numeric(as.character(Only\_clean\_reports$'WFF+MSA Impact on Company Behaviour1'))

**##"WFF+MSA policy (revised)"**

bigvector=c()  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])  
{ifelse(  
 grepl("policies to combat modern slavery",sent, ignore.case = TRUE)|  
grepl("organisational policies",sent, ignore.case = TRUE)|  
grepl("modern slavery policies",sent, ignore.case = TRUE)|   
grepl(" comply with laws and company’s policies",sent, ignore.case = TRUE)|   
grepl("Prohibit use of forced labour",sent, ignore.case = TRUE)|   
grepl("Code of conduct",sent, ignore.case = TRUE)|   
grepl("supplier code",sent, ignore.case = TRUE)|   
grepl("Contracts include clauses",sent, ignore.case = TRUE)|   
grepl("Suppliers produce their own statement",sent, ignore.case = TRUE)|   
grepl("Suppliers respect labour rights",sent, ignore.case = TRUE)|   
grepl("Prohibit charging of recruitment fees to employee ",sent, ignore.case = TRUE)|   
grepl("protect migrant workers",sent, ignore.case = TRUE)|   
grepl("Policy",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA policy (revised)'=bigvector

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])   
{ifelse(  
grepl("policies to combat modern slavery",sent, ignore.case = TRUE)|  
grepl("organisational policies",sent, ignore.case = TRUE)|  
grepl("modern slavery policies",sent, ignore.case = TRUE)|   
grepl(" comply with laws and company’s policies",sent, ignore.case = TRUE)|   
grepl("Prohibit use of forced labour",sent, ignore.case = TRUE)|   
grepl("Code of conduct",sent, ignore.case = TRUE)|   
grepl("supplier code",sent, ignore.case = TRUE)|   
grepl("Contracts include clauses",sent, ignore.case = TRUE)|   
grepl("Suppliers produce their own statement",sent, ignore.case = TRUE)|   
grepl("Suppliers respect labour rights",sent, ignore.case = TRUE)|   
grepl("Prohibit charging of recruitment fees to employee ",sent, ignore.case = TRUE)|   
grepl("protect migrant workers",sent, ignore.case = TRUE)|   
grepl("Policy",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA policy (revised)1'=bigvector  
Only\_clean\_reports$'WFF+MSA policy (revised)1' <- as.numeric(as.character(Only\_clean\_reports$'WFF+MSA policy (revised)1'))

**##"BHRRC+MSS Approval"**

bigvector=c()  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])  
{ifelse(  
 grepl("approve the financial statement",sent, ignore.case = TRUE) |   
 grepl("board approval",sent, ignore.case = TRUE)|   
 grepl("approved by the board",sent, ignore.case = TRUE)|   
 grepl("approved by the boards",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s board",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s managing",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s chief",sent, ignore.case = TRUE)|   
 grepl("approved by our boards",sent, ignore.case = TRUE)|   
 grepl("approved by our directors",sent, ignore.case = TRUE)|   
 grepl("approved by our managing",sent, ignore.case = TRUE)|   
 grepl("approved by our managing director",sent, ignore.case = TRUE)|  
 grepl("approved by board",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'BHRRC+MSS Approval'=bigvector

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])   
{ifelse(  
grepl("approve the financial statement",sent, ignore.case = TRUE) |   
 grepl("board approval",sent, ignore.case = TRUE)|   
 grepl("approved by the board",sent, ignore.case = TRUE)|   
 grepl("approved by the boards",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s board",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s managing",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s chief",sent, ignore.case = TRUE)|   
 grepl("approved by our boards",sent, ignore.case = TRUE)|   
 grepl("approved by our directors",sent, ignore.case = TRUE)|   
 grepl("approved by our managing",sent, ignore.case = TRUE)|   
 grepl("approved by our managing director",sent, ignore.case = TRUE)|  
 grepl("approved by board",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'BHRRC+MSS Approval1'=bigvector  
Only\_clean\_reports$'BHRRC+MSS Approval1' <- as.numeric(as.character(Only\_clean\_reports$'BHRRC+MSS Approval1'))

**##"BHRRC+MSA Statement Signed"**

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl("signed on behalf of",sent, ignore.case = TRUE)|

grepl("signed on its behalf",sent, ignore.case = TRUE)|

grepl("signed on their behalf",sent, ignore.case = TRUE)|

grepl("signed by the",sent, ignore.case = TRUE)|

grepl("is signed below",sent, ignore.case = TRUE)|

grepl("signed by the",sent, ignore.case = TRUE)|

grepl("signed by a",sent, ignore.case = TRUE)|

grepl("signature",sent, ignore.case = TRUE)

,vector<-sent,x<-0)};

{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}

Only\_clean\_reports$'BHRRC+MSA Statement Signed'=bigvector

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl("signed on behalf of",sent, ignore.case = TRUE)|

grepl("signed on its behalf",sent, ignore.case = TRUE)|

grepl("signed on their behalf",sent, ignore.case = TRUE)|

grepl("signed by the",sent, ignore.case = TRUE)|

grepl("is signed below",sent, ignore.case = TRUE)|

grepl("signed by the",sent, ignore.case = TRUE)|

grepl("signed by a",sent, ignore.case = TRUE)|

grepl("signature",sent, ignore.case = TRUE)

,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'BHRRC+MSA Statement Signed1'=bigvector  
  
Only\_clean\_reports$`BHRRC+MSA Statement Signed1` <- as.numeric(as.character(Only\_clean\_reports$`BHRRC+MSA Statement Signed1`))

**##"WFF+MSA supply chain disclosure"**

bigvector=c()  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])  
{ifelse(  
 grepl("map our supply",sent, ignore.case = TRUE)|   
grepl( "map supply",sent, ignore.case = TRUE)|   
grepl(" map factory",sent, ignore.case = TRUE)|   
grepl("map our factory",sent, ignore.case = TRUE)|   
grepl(" map our suppliers",sent, ignore.case = TRUE)|   
grepl(" identify the suppliers",sent, ignore.case = TRUE)|   
grepl("identify suppliers",sent, ignore.case = TRUE)|

grepl("disclose supply",sent, ignore.case = TRUE)|   
grepl("disclosing supply",sent, ignore.case = TRUE)|   
grepl(" we operate in",sent, ignore.case = TRUE)|   
grepl(" list of countries",sent, ignore.case = TRUE)|   
grepl("list of regions",sent, ignore.case = TRUE)|   
grepl(" supply chain is based",sent, ignore.case = TRUE)|   
grepl("disclose suppliers",sent, ignore.case = TRUE)|   
grepl("disclosing suppliers",sent, ignore.case = TRUE)|   
grepl("source country of origin of our suppliers",sent, ignore.case = TRUE)|   
grepl("source products from",sent, ignore.case = TRUE)|   
grepl("suppliers based in",sent, ignore.case = TRUE)|   
grepl("source from",sent, ignore.case = TRUE)|   
grepl("diagram below",sent, ignore.case = TRUE)|   
grepl("manufactures in",sent, ignore.case = TRUE)|   
grepl("who are based in ",sent, ignore.case = TRUE)|   
grepl("import from",sent, ignore.case = TRUE)|   
grepl("sourcing hub ",sent, ignore.case = TRUE)|   
grepl("supplier sites based",sent, ignore.case = TRUE)|   
grepl("suppliers sites based",sent, ignore.case = TRUE)|   
grepl("businesses sites based",sent, ignore.case = TRUE)|   
grepl("operations are predominantly located",sent, ignore.case = TRUE)|   
grepl("operations are located in",sent, ignore.case = TRUE)|   
grepl("from manufacturers based in",sent, ignore.case = TRUE)|   
grepl("we source from",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA supply chain disclosure'=bigvector

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])   
{ifelse(  
 grepl("map our supply",sent, ignore.case = TRUE)|   
grepl( "map supply",sent, ignore.case = TRUE)|   
grepl(" map factory",sent, ignore.case = TRUE)|   
grepl("map our factory",sent, ignore.case = TRUE)|   
grepl(" map our suppliers",sent, ignore.case = TRUE)|   
grepl(" identify the suppliers",sent, ignore.case = TRUE)|   
grepl("identify suppliers",sent, ignore.case = TRUE)|

grepl("disclose supply",sent, ignore.case = TRUE)|   
grepl("disclosing supply",sent, ignore.case = TRUE)|   
grepl(" we operate in",sent, ignore.case = TRUE)|   
grepl(" list of countries",sent, ignore.case = TRUE)|   
grepl("list of regions",sent, ignore.case = TRUE)|   
grepl(" supply chain is based",sent, ignore.case = TRUE)|   
grepl("disclose suppliers",sent, ignore.case = TRUE)|   
grepl("disclosing suppliers",sent, ignore.case = TRUE)|   
grepl("source country of origin of our suppliers",sent, ignore.case = TRUE)|   
grepl("source products from",sent, ignore.case = TRUE)|   
grepl("suppliers based in",sent, ignore.case = TRUE)|   
grepl("source from",sent, ignore.case = TRUE)|   
grepl("diagram below",sent, ignore.case = TRUE)|   
grepl("manufactures in",sent, ignore.case = TRUE)|   
grepl("who are based in ",sent, ignore.case = TRUE)|   
grepl("import from",sent, ignore.case = TRUE)|   
grepl("sourcing hub ",sent, ignore.case = TRUE)|   
grepl("supplier sites based",sent, ignore.case = TRUE)|   
grepl("suppliers sites based",sent, ignore.case = TRUE)|   
grepl("businesses sites based",sent, ignore.case = TRUE)|   
grepl("operations are predominantly located",sent, ignore.case = TRUE)|   
grepl("operations are located in",sent, ignore.case = TRUE)|   
grepl("from manufacturers based in",sent, ignore.case = TRUE)|   
grepl("we source from",sent, ignore.case = TRUE) ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA supply chain disclosure1'=bigvector  
Only\_clean\_reports$'WFF+MSA supply chain disclosure1' <- as.numeric(as.character(Only\_clean\_reports$'WFF+MSA supply chain disclosure1'))

**##WFF+MSA incidents remediation (revised)**

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl(" if a supplier is found to be",sent, ignore.case = TRUE)|   
grepl("providing remediation direct to the worker affected",sent, ignore.case = TRUE)|   
grepl("backpaymen",sent, ignore.case = TRUE)|   
grepl("support to prosecute",sent, ignore.case = TRUE)|   
grepl("informing senior management",sent, ignore.case = TRUE)|   
grepl(" respond to instances of",sent, ignore.case = TRUE)|   
grepl("instigating corrective action plans",sent, ignore.case = TRUE)|   
grepl("cancelling the contracts of suppliers",sent, ignore.case = TRUE)|   
grepl("supports the supplier to respond",sent, ignore.case = TRUE)|   
grepl("corrective action plan",sent, ignore.case = TRUE)|   
grepl("punitive action",sent, ignore.case = TRUE)|   
grepl("developing a remediation policy",sent, ignore.case = TRUE)|   
grepl("planning to implement a remediation policy",sent, ignore.case = TRUE)|   
grepl("contract termination",sent, ignore.case = TRUE)|   
grepl("termination of the",sent, ignore.case = TRUE)|   
grepl("termination for failure",sent, ignore.case = TRUE)|   
grepl("will not deal with any supplier if",sent, ignore.case = TRUE)|   
grepl("reserves the right to terminate",sent, ignore.case = TRUE)|   
grepl("to terminate",sent, ignore.case = TRUE)|   
grepl("cessation of",sent, ignore.case = TRUE)|   
grepl("cease",sent, ignore.case = TRUE)|   
grepl("curtailment",sent, ignore.case = TRUE)|   
grepl("knowingly involved ",sent, ignore.case = TRUE)|   
grepl("cancel contracts",sent, ignore.case = TRUE)|   
grepl("cancel contract",sent, ignore.case = TRUE)|   
grepl("remedial action",sent, ignore.case = TRUE)|   
grepl("appropriate steps to",sent, ignore.case = TRUE)|   
grepl("continuously improve our processes to fight",sent, ignore.case = TRUE)|   
grepl("we require evidence",sent, ignore.case = TRUE)|   
grepl("action plans",sent, ignore.case = TRUE)

,vector<-sent,x<-0)};

{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}

Only\_clean\_reports$'WFF+MSA incidents remediation (revised)'=bigvector

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl(" if a supplier is found to be",sent, ignore.case = TRUE)|   
grepl("providing remediation direct to the worker affected",sent, ignore.case = TRUE)|   
grepl("backpaymen",sent, ignore.case = TRUE)|   
grepl("support to prosecute",sent, ignore.case = TRUE)|   
grepl("informing senior management",sent, ignore.case = TRUE)|   
grepl("respond to instances of",sent, ignore.case = TRUE)|   
grepl("instigating corrective action plans",sent, ignore.case = TRUE)|   
grepl("cancelling the contracts of suppliers",sent, ignore.case = TRUE)|   
grepl("supports the supplier to respond",sent, ignore.case = TRUE)|   
grepl("corrective action plan",sent, ignore.case = TRUE)|   
grepl("punitive action",sent, ignore.case = TRUE)|   
grepl("developing a remediation policy",sent, ignore.case = TRUE)|   
grepl("planning to implement a remediation policy",sent, ignore.case = TRUE)|   
grepl("contract termination",sent, ignore.case = TRUE)|   
grepl("termination of the",sent, ignore.case = TRUE)|   
grepl("termination for failure",sent, ignore.case = TRUE)|   
grepl("will not deal with any supplier if",sent, ignore.case = TRUE)|   
grepl("reserves the right to terminate",sent, ignore.case = TRUE)|   
grepl("to terminate",sent, ignore.case = TRUE)|   
grepl("cessation of",sent, ignore.case = TRUE)|   
grepl("cease",sent, ignore.case = TRUE)|   
grepl("curtailment",sent, ignore.case = TRUE)|   
grepl("knowingly involved ",sent, ignore.case = TRUE)|   
grepl("cancel contracts",sent, ignore.case = TRUE)|   
grepl("cancel contract",sent, ignore.case = TRUE)|   
grepl("remedial action",sent, ignore.case = TRUE)|   
grepl("appropriate steps to",sent, ignore.case = TRUE)|   
grepl("continuously improve our processes to fight",sent, ignore.case = TRUE)|   
grepl("we require evidence",sent, ignore.case = TRUE)|   
grepl("action plans",sent, ignore.case = TRUE)

,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA incidents remediation (revised)1'=bigvector  
  
Only\_clean\_reports$`WFF+MSA incidents remediation (revised)1` <- as.numeric(as.character(Only\_clean\_reports$`WFF+MSA incidents remediation (revised)1`))

**## "WFF+MSA whistleblowing mechanism (revised)"**

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl("hotline",sent, ignore.case = TRUE)|   
grepl("grievances",sent, ignore.case = TRUE)|   
grepl("suspected incidents",sent, ignore.case = TRUE)|   
grepl("reporting line",sent, ignore.case = TRUE)|   
grepl("anonymous reporting",sent, ignore.case = TRUE)|   
grepl("workers can call",sent, ignore.case = TRUE)|   
grepl("whistleblower",sent, ignore.case = TRUE)|   
grepl("report modern slavery cases",sent, ignore.case = TRUE)|   
grepl("focal point",sent, ignore.case = TRUE)|   
grepl("report any suspic",sent, ignore.case = TRUE)|   
grepl("identify and report",sent, ignore.case = TRUE)|   
grepl("appropriate steps to",sent, ignore.case = TRUE)|  
grepl("we encourage",sent, ignore.case = TRUE)|  
grepl("self-assessment questionnaire",sent, ignore.case = TRUE)|  
grepl("confidential",sent, ignore.case = TRUE)|  
grepl("mechanism for reporting",sent, ignore.case = TRUE)|  
grepl("mechanism to",sent, ignore.case = TRUE)|  
grepl("whistleblowing",sent, ignore.case = TRUE)|  
grepl("whistleblower",sent, ignore.case = TRUE)|   
grepl("whistleblowers",sent, ignore.case = TRUE)|   
grepl("protection",sent, ignore.case = TRUE)

,vector<-sent,x<-0)};

{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}

Only\_clean\_reports$'WFF+MSA whistleblowing mechanism (revised)'=bigvector

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl("hotline",sent, ignore.case = TRUE)|   
grepl("grievances",sent, ignore.case = TRUE)|   
grepl("suspected incidents",sent, ignore.case = TRUE)|   
grepl("reporting line",sent, ignore.case = TRUE)|   
grepl("anonymous reporting",sent, ignore.case = TRUE)|   
grepl("workers can call",sent, ignore.case = TRUE)|   
grepl("whistleblower",sent, ignore.case = TRUE)|   
grepl("report modern slavery cases",sent, ignore.case = TRUE)|   
grepl("focal point",sent, ignore.case = TRUE)|   
grepl("report any suspic",sent, ignore.case = TRUE)|   
grepl("identify and report",sent, ignore.case = TRUE)|   
grepl("appropriate steps to",sent, ignore.case = TRUE)|  
grepl("we encourage",sent, ignore.case = TRUE)|  
grepl("self-assessment questionnaire",sent, ignore.case = TRUE)|  
grepl("confidential",sent, ignore.case = TRUE)|  
grepl("mechanism for reporting",sent, ignore.case = TRUE)|  
grepl("mechanism to",sent, ignore.case = TRUE)|  
grepl("whistleblowing",sent, ignore.case = TRUE)|  
grepl("whistleblower",sent, ignore.case = TRUE)|   
grepl("whistleblowers",sent, ignore.case = TRUE)|   
grepl("protection",sent, ignore.case = TRUE)

,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA whistleblowing mechanism (revised)1'=bigvector  
  
Only\_clean\_reports$`WFF+MSA whistleblowing mechanism (revised)1` <- as.numeric(as.character(Only\_clean\_reports$`WFF+MSA whistleblowing mechanism (revised)1`))

**##"WFF+MSA training (revised)"**

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl("provide training",sent, ignore.case = TRUE)|  
grepl("training",sent, ignore.case = TRUE)|  
grepl("course on modern slavery",sent, ignore.case = TRUE)|  
grepl("training programme",sent, ignore.case = TRUE)|  
grepl("Procurement",sent, ignore.case = TRUE)|   
grepl("purchasing",sent, ignore.case = TRUE)|   
grepl("Recruitment",sent, ignore.case = TRUE)|   
grepl("HR",sent, ignore.case = TRUE)|   
grepl("training provided",sent, ignore.case = TRUE)|   
grepl("Recruitment",sent, ignore.case = TRUE)|   
grepl(" human resources",sent, ignore.case = TRUE)|   
grepl("training provider",sent, ignore.case = TRUE)|   
grepl("training provider",sent, ignore.case = TRUE)|   
grepl("training provider",sent, ignore.case = TRUE)|   
grepl("training provider",sent, ignore.case = TRUE)

,vector<-sent,x<-0)};

{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}

Only\_clean\_reports$'WFF+MSA training (revised)'=bigvector

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl("provide training",sent, ignore.case = TRUE)|  
grepl("training",sent, ignore.case = TRUE)|  
grepl("course on modern slavery",sent, ignore.case = TRUE)|  
grepl("training programme",sent, ignore.case = TRUE)|  
grepl("Procurement",sent, ignore.case = TRUE)|   
grepl("purchasing",sent, ignore.case = TRUE)|   
grepl("Recruitment",sent, ignore.case = TRUE)|   
grepl("HR",sent, ignore.case = TRUE)|   
grepl("training provided",sent, ignore.case = TRUE)|   
grepl("Recruitment",sent, ignore.case = TRUE)|   
grepl(" human resources",sent, ignore.case = TRUE)|   
grepl("training provider",sent, ignore.case = TRUE)|   
grepl("training provider",sent, ignore.case = TRUE)|   
grepl("training provider",sent, ignore.case = TRUE)|   
grepl("training provider",sent, ignore.case = TRUE)

,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA training (revised)1'=bigvector  
  
Only\_clean\_reports$`WFF+MSA training (revised)1` <- as.numeric(as.character(Only\_clean\_reports$`WFF+MSA training (revised)1`))

**##"WFF+MSA Performance Indicators"**

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl("kpi",sent, ignore.case = TRUE)|   
grepl("kpis",sent, ignore.case = TRUE)|   
grepl("key Performance Indicator",sent, ignore.case = TRUE)|   
grepl("key Performance Indicators",sent, ignore.case = TRUE)|   
grepl("measure impact",sent, ignore.case = TRUE)|   
grepl("action taken",sent, ignore.case = TRUE)|   
grepl("combat modern slavery",sent, ignore.case = TRUE)|   
grepl("measuring the impact of training",sent, ignore.case = TRUE)|   
grepl(" number of audits",sent, ignore.case = TRUE)|   
grepl("on site visits",sent, ignore.case = TRUE)|  
grepl(" suppliers questionnaires",sent, ignore.case = TRUE)|  
grepl("monitoring mechanisms",sent, ignore.case = TRUE)|  
grepl("assess the risk",sent, ignore.case = TRUE)|  
grepl("review our",sent, ignore.case = TRUE)|  
grepl("measure our",sent, ignore.case = TRUE)|  
grepl("review and",sent, ignore.case = TRUE) ,vector<-sent,x<-0)};

{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}

Only\_clean\_reports$'WFF+MSA Performance Indicators'=bigvector

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl("kpi",sent, ignore.case = TRUE)|   
grepl("kpis",sent, ignore.case = TRUE)|   
grepl("key Performance Indicator",sent, ignore.case = TRUE)|   
grepl("key Performance Indicators",sent, ignore.case = TRUE)|   
grepl("measure impact",sent, ignore.case = TRUE)|   
grepl("action taken",sent, ignore.case = TRUE)|   
grepl("combat modern slavery",sent, ignore.case = TRUE)|   
grepl("measuring the impact of training",sent, ignore.case = TRUE)|   
grepl(" number of audits",sent, ignore.case = TRUE)|   
grepl("on site visits",sent, ignore.case = TRUE)|  
grepl(" suppliers questionnaires",sent, ignore.case = TRUE)|  
grepl("monitoring mechanisms",sent, ignore.case = TRUE)|  
grepl("assess the risk",sent, ignore.case = TRUE)|  
grepl("review our",sent, ignore.case = TRUE)|  
grepl("measure our",sent, ignore.case = TRUE)|  
grepl("review and",sent, ignore.case = TRUE) ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA Performance Indicators1'=bigvector  
  
Only\_clean\_reports$`WFF+MSA Performance Indicators1` <- as.numeric(as.character(Only\_clean\_reports$`WFF+MSA Performance Indicators1`))

**##"WFF+MSA Business Performance Indicators"**

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl("kpi",sent, ignore.case = TRUE)|   
grepl("kpis",sent, ignore.case = TRUE)|   
grepl("Performance Indicator",sent, ignore.case = TRUE)|   
grepl("Performance Indicators",sent, ignore.case = TRUE)|   
grepl("turn-around time",sent, ignore.case = TRUE)|   
grepl("cost of materials",sent, ignore.case = TRUE)|   
grepl("efficiency in production",sent, ignore.case = TRUE)|   
grepl("cheapest goods",sent, ignore.case = TRUE)|   
grepl("cheape goods",sent, ignore.case = TRUE)|   
grepl("sourcing the cheapest goods in the shortest amount of time",sent, ignore.case = TRUE)

,vector<-sent,x<-0)};

{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}

Only\_clean\_reports$'WFF+MSA Business Performance Indicators'=bigvector

bigvector=c()

for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(

grepl("kpi",sent, ignore.case = TRUE)|   
grepl("kpis",sent, ignore.case = TRUE)|   
grepl("Performance Indicator",sent, ignore.case = TRUE)|   
grepl("Performance Indicators",sent, ignore.case = TRUE)|   
grepl("turn-around time",sent, ignore.case = TRUE)|   
grepl("cost of materials",sent, ignore.case = TRUE)|   
grepl("efficiency in production",sent, ignore.case = TRUE)|   
grepl("cheapest goods",sent, ignore.case = TRUE)|   
grepl("cheape goods",sent, ignore.case = TRUE)|   
grepl("sourcing the cheapest goods in the shortest amount of time",sent, ignore.case = TRUE)

,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA Business Performance Indicators1'=bigvector  
  
Only\_clean\_reports$`WFF+MSA Business Performance Indicators1` <- as.numeric(as.character(Only\_clean\_reports$`WFF+MSA Business Performance Indicators1`))

**##"WFF+MSA incidents identified"**

bigvector=c()  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("incident identified",sent, ignore.case = TRUE)|

grepl("incident was identified",sent, ignore.case = TRUE)|

grepl("incidents wereidentified",sent, ignore.case = TRUE)|  
grepl("issues found",sent, ignore.case = TRUE)|  
grepl("issues were identified",sent, ignore.case = TRUE)|  
grepl("issue was identified",sent, ignore.case = TRUE)|  
grepl("found isssue",sent, ignore.case = TRUE)|

grepl("found incident",sent, ignore.case = TRUE)|   
grepl("issue found",sent, ignore.case = TRUE)|

grepl("issues found ",sent, ignore.case = TRUE)

,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA incidents identified'=bigvector  
  
  
bigvector=c()  
  
  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("incident identified",sent, ignore.case = TRUE)|

grepl("incident was identified",sent, ignore.case = TRUE)|

grepl("incidents wereidentified",sent, ignore.case = TRUE)|  
grepl("issues found",sent, ignore.case = TRUE)|  
grepl("issues were identified",sent, ignore.case = TRUE)|  
grepl("issue was identified",sent, ignore.case = TRUE)|  
grepl("found isssue",sent, ignore.case = TRUE)|

grepl("found incident",sent, ignore.case = TRUE)|   
grepl("issue found",sent, ignore.case = TRUE)|

grepl("issues found ",sent, ignore.case = TRUE)

,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA incidents identified1'=bigvector  
  
Only\_clean\_reports$`WFF+MSA incidents identified1` <- as.numeric(as.character(Only\_clean\_reports$`WFF+MSA incidents identified1`))

**##"WFF+MSA Identification of risks"**

bigvector=c()  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("identify risk",sent, ignore.case = TRUE)|  
grepl("risk profile",sent, ignore.case = TRUE)|  
grepl(" risks were identified",sent, ignore.case = TRUE)|  
grepl("risk was identified",sent, ignore.case = TRUE)|  
grepl("found risk",sent, ignore.case = TRUE)|   
grepl("have identified risk",sent, ignore.case = TRUE)|   
grepl("we have identified risk",sent, ignore.case = TRUE)

,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA Identification of risks'=bigvector  
  
  
bigvector=c()  
  
  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("identify risk",sent, ignore.case = TRUE)|  
grepl("risk profile",sent, ignore.case = TRUE)|  
grepl(" risks were identified",sent, ignore.case = TRUE)|  
grepl("risk was identified",sent, ignore.case = TRUE)|  
grepl("found risk",sent, ignore.case = TRUE)|   
grepl("have identified risk",sent, ignore.case = TRUE)|   
grepl("we have identified risk",sent, ignore.case = TRUE)  
,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA Identification of risks1'=bigvector  
  
Only\_clean\_reports$`WFF+MSA Identification of risks1` <- as.numeric(as.character(Only\_clean\_reports$`WFF+MSA Identification of risks1`))

**##"WFF+MSA risk management (revised)"**

bigvector=c()  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("due diligence",sent, ignore.case = TRUE)|  
grepl("planning to implement",sent, ignore.case = TRUE)|  
grepl("continuous improvement programs",sent, ignore.case = TRUE)|  
grepl("audit of suppliers",sent, ignore.case = TRUE)|  
grepl("continuously engaging with suppliers",sent, ignore.case = TRUE)|   
grepl("on-site visits",sent, ignore.case = TRUE)|   
grepl("audits of suppliers",sent, ignore.case = TRUE)|   
grepl("audits",sent, ignore.case = TRUE)|   
grepl("audit",sent, ignore.case = TRUE)|   
grepl("monitor",sent, ignore.case = TRUE)|   
grepl("third party",sent, ignore.case = TRUE)|   
grepl("verif",sent, ignore.case = TRUE)

,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA risk management (revised)'=bigvector  
  
  
bigvector=c()  
  
  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("due diligence",sent, ignore.case = TRUE)|  
grepl("planning to implement",sent, ignore.case = TRUE)|  
grepl("continuous improvement programs",sent, ignore.case = TRUE)|  
grepl("audit of suppliers",sent, ignore.case = TRUE)|  
grepl("continuously engaging with suppliers",sent, ignore.case = TRUE)|   
grepl("on-site visits",sent, ignore.case = TRUE)|   
grepl("audits of suppliers",sent, ignore.case = TRUE)|   
grepl("audits",sent, ignore.case = TRUE)|   
grepl("audit",sent, ignore.case = TRUE)|   
grepl("monitor",sent, ignore.case = TRUE)|   
grepl("third party",sent, ignore.case = TRUE)|   
grepl("verif",sent, ignore.case = TRUE)  
,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA risk management (revised)1'=bigvector  
  
Only\_clean\_reports$`WFF+MSA risk management (revised)1` <- as.numeric(as.character(Only\_clean\_reports$`WFF+MSA risk management (revised)1`))

## **"WFF+MSA risk assessment"**

bigvector=c()  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("assess risk",sent, ignore.case = TRUE)|  
grepl("risk-based questionnaires",sent, ignore.case = TRUE)|  
grepl("risk management database",sent, ignore.case = TRUE)|  
grepl("risk management tool",sent, ignore.case = TRUE)|  
grepl("Maplecrofts",sent, ignore.case = TRUE)|   
grepl("Sedex",sent, ignore.case = TRUE)|   
grepl("risk matrix",sent, ignore.case = TRUE)|   
grepl("conducting research",sent, ignore.case = TRUE)|   
grepl("risk assessment",sent, ignore.case = TRUE)|   
grepl("assessing",sent, ignore.case = TRUE)

,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA risk assessment'=bigvector  
  
  
bigvector=c()  
  
  
for (i in 1:length(Only\_clean\_reports$text)) { doc = Only\_clean\_reports$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("assess risk",sent, ignore.case = TRUE)|  
grepl("risk-based questionnaires",sent, ignore.case = TRUE)|  
grepl("risk management database",sent, ignore.case = TRUE)|  
grepl("risk management tool",sent, ignore.case = TRUE)|  
grepl("Maplecrofts",sent, ignore.case = TRUE)|   
grepl("Sedex",sent, ignore.case = TRUE)|   
grepl("risk matrix",sent, ignore.case = TRUE)|   
grepl("conducting research",sent, ignore.case = TRUE)|   
grepl("risk assessment",sent, ignore.case = TRUE)|   
grepl("assessing",sent, ignore.case = TRUE)  
,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Only\_clean\_reports$'WFF+MSA risk assessment1'=bigvector  
  
Only\_clean\_reports$`WFF+MSA risk assessment1` <- as.numeric(as.character(Only\_clean\_reports$`WFF+MSA risk assessment1`))

**"BHRRC+MSA Statement Homepage Link"**

Only\_clean\_reports$'BHRRC+MSA Statement Homepage Link'<- 1  
  
Only\_clean\_reports$`BHRRC+MSA##WFF+MSA Impact on Company Behaviour"   
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])  
{ifelse(  
grepl("Updating the company’s code of conduct",sent, ignore.case = TRUE)|   
grepl("Updating the supplier code",sent, ignore.case = TRUE)|   
grepl("new modern slavery company policy ",sent, ignore.case = TRUE)|   
grepl("Updating supplier contracts",sent, ignore.case = TRUE)|   
grepl("new training programme",sent, ignore.case = TRUE)|   
grepl(" new key performance indicators",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA Impact on Company Behaviour'=bigvector  
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])   
{ifelse(  
grepl("Updating the company’s code of conduct",sent, ignore.case = TRUE)|   
grepl("Updating the supplier code",sent, ignore.case = TRUE)|   
grepl("new modern slavery company policy ",sent, ignore.case = TRUE)|   
grepl("Updating supplier contracts",sent, ignore.case = TRUE)|   
grepl("new training programme",sent, ignore.case = TRUE)|   
grepl(" new key performance indicators",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA Impact on Company Behaviour1'=bigvector  
Total$'WFF+MSA Impact on Company Behaviour1' <- as.numeric(as.character(Total$'WFF+MSA Impact on Company Behaviour1'))  
  
  
##"WFF+MSA policy (revised)"   
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])  
{ifelse(  
 grepl("policies to combat modern slavery",sent, ignore.case = TRUE)|  
grepl("organisational policies",sent, ignore.case = TRUE)|  
grepl("modern slavery policies",sent, ignore.case = TRUE)|   
grepl(" comply with laws and company’s policies",sent, ignore.case = TRUE)|   
grepl("Prohibit use of forced labour",sent, ignore.case = TRUE)|   
grepl("Code of conduct",sent, ignore.case = TRUE)|   
grepl("supplier code",sent, ignore.case = TRUE)|   
grepl("Contracts include clauses",sent, ignore.case = TRUE)|   
grepl("Suppliers produce their own statement",sent, ignore.case = TRUE)|   
grepl("Suppliers respect labour rights",sent, ignore.case = TRUE)|   
grepl("Prohibit charging of recruitment fees to employee ",sent, ignore.case = TRUE)|   
grepl("protect migrant workers",sent, ignore.case = TRUE)|   
grepl("Policy",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA policy (revised)'=bigvector  
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])   
{ifelse(  
grepl("policies to combat modern slavery",sent, ignore.case = TRUE)|  
grepl("organisational policies",sent, ignore.case = TRUE)|  
grepl("modern slavery policies",sent, ignore.case = TRUE)|   
grepl(" comply with laws and company’s policies",sent, ignore.case = TRUE)|   
grepl("Prohibit use of forced labour",sent, ignore.case = TRUE)|   
grepl("Code of conduct",sent, ignore.case = TRUE)|   
grepl("supplier code",sent, ignore.case = TRUE)|   
grepl("Contracts include clauses",sent, ignore.case = TRUE)|   
grepl("Suppliers produce their own statement",sent, ignore.case = TRUE)|   
grepl("Suppliers respect labour rights",sent, ignore.case = TRUE)|   
grepl("Prohibit charging of recruitment fees to employee ",sent, ignore.case = TRUE)|   
grepl("protect migrant workers",sent, ignore.case = TRUE)|   
grepl("Policy",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA policy (revised)1'=bigvector  
Total$'WFF+MSA policy (revised)1' <- as.numeric(as.character(Total$'WFF+MSA policy (revised)1'))  
  
  
##"BHRRC+MSS Approval"   
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])  
{ifelse(  
 grepl("approve the financial statement",sent, ignore.case = TRUE) |   
 grepl("board approval",sent, ignore.case = TRUE)|   
 grepl("approved by the board",sent, ignore.case = TRUE)|   
 grepl("approved by the boards",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s board",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s managing",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s chief",sent, ignore.case = TRUE)|   
 grepl("approved by our boards",sent, ignore.case = TRUE)|   
 grepl("approved by our directors",sent, ignore.case = TRUE)|   
 grepl("approved by our managing",sent, ignore.case = TRUE)|   
 grepl("approved by our managing director",sent, ignore.case = TRUE)|  
 grepl("approved by board",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'BHRRC+MSS Approval'=bigvector  
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])   
{ifelse(  
grepl("approve the financial statement",sent, ignore.case = TRUE) |   
 grepl("board approval",sent, ignore.case = TRUE)|   
 grepl("approved by the board",sent, ignore.case = TRUE)|   
 grepl("approved by the boards",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s board",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s managing",sent, ignore.case = TRUE)|   
 grepl("approved by the company’s chief",sent, ignore.case = TRUE)|   
 grepl("approved by our boards",sent, ignore.case = TRUE)|   
 grepl("approved by our directors",sent, ignore.case = TRUE)|   
 grepl("approved by our managing",sent, ignore.case = TRUE)|   
 grepl("approved by our managing director",sent, ignore.case = TRUE)|  
 grepl("approved by board",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'BHRRC+MSS Approval1'=bigvector  
Total$'BHRRC+MSS Approval1' <- as.numeric(as.character(Total$'BHRRC+MSS Approval1'))  
  
  
  
##"BHRRC+MSA Statement Signed"   
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("signed on behalf of",sent, ignore.case = TRUE)|   
grepl("signed on its behalf",sent, ignore.case = TRUE)|   
grepl("signed on their behalf",sent, ignore.case = TRUE)|   
grepl("signed by the",sent, ignore.case = TRUE)|   
grepl("is signed below",sent, ignore.case = TRUE)|   
grepl("signed by the",sent, ignore.case = TRUE)|   
grepl("signed by a",sent, ignore.case = TRUE)|   
grepl("signature",sent, ignore.case = TRUE)  
 ,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'BHRRC+MSA Statement Signed'=bigvector  
  
bigvector=c()  
  
  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("signed on behalf of",sent, ignore.case = TRUE)|   
grepl("signed on its behalf",sent, ignore.case = TRUE)|   
grepl("signed on their behalf",sent, ignore.case = TRUE)|   
grepl("signed by the",sent, ignore.case = TRUE)|   
grepl("is signed below",sent, ignore.case = TRUE)|   
grepl("signed by the",sent, ignore.case = TRUE)|   
grepl("signed by a",sent, ignore.case = TRUE)|   
grepl("signature",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'BHRRC+MSA Statement Signed1'=bigvector  
  
Total$`BHRRC+MSA Statement Signed1` <- as.numeric(as.character(Total$`BHRRC+MSA Statement Signed1`))  
  
  
##"WFF+MSA supply chain disclosure"   
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])  
{ifelse(  
 grepl("map our supply",sent, ignore.case = TRUE)|   
grepl( "map supply",sent, ignore.case = TRUE)|   
grepl(" map factory",sent, ignore.case = TRUE)|   
grepl("map our factory",sent, ignore.case = TRUE)|   
grepl(" map our suppliers",sent, ignore.case = TRUE)|   
grepl(" identify the suppliers",sent, ignore.case = TRUE)|   
grepl("identify suppliers",sent, ignore.case = TRUE)|   
grepl("disclose supply",sent, ignore.case = TRUE)|   
grepl("disclosing supply",sent, ignore.case = TRUE)|   
grepl(" we operate in",sent, ignore.case = TRUE)|   
grepl(" list of countries",sent, ignore.case = TRUE)|   
grepl("list of regions",sent, ignore.case = TRUE)|   
grepl(" supply chain is based",sent, ignore.case = TRUE)|   
grepl("disclose suppliers",sent, ignore.case = TRUE)|   
grepl("disclosing suppliers",sent, ignore.case = TRUE)|   
grepl("source country of origin of our suppliers",sent, ignore.case = TRUE)|   
grepl("source products from",sent, ignore.case = TRUE)|   
grepl("suppliers based in",sent, ignore.case = TRUE)|   
grepl("source from",sent, ignore.case = TRUE)|   
grepl("diagram below",sent, ignore.case = TRUE)|   
grepl("manufactures in",sent, ignore.case = TRUE)|   
grepl("who are based in ",sent, ignore.case = TRUE)|   
grepl("import from",sent, ignore.case = TRUE)|   
grepl("sourcing hub ",sent, ignore.case = TRUE)|   
grepl("supplier sites based",sent, ignore.case = TRUE)|   
grepl("suppliers sites based",sent, ignore.case = TRUE)|   
grepl("businesses sites based",sent, ignore.case = TRUE)|   
grepl("operations are predominantly located",sent, ignore.case = TRUE)|   
grepl("operations are located in",sent, ignore.case = TRUE)|   
grepl("from manufacturers based in",sent, ignore.case = TRUE)|   
grepl("we source from",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA supply chain disclosure'=bigvector  
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]])   
{ifelse(  
 grepl("map our supply",sent, ignore.case = TRUE)|   
grepl( "map supply",sent, ignore.case = TRUE)|   
grepl(" map factory",sent, ignore.case = TRUE)|   
grepl("map our factory",sent, ignore.case = TRUE)|   
grepl(" map our suppliers",sent, ignore.case = TRUE)|   
grepl(" identify the suppliers",sent, ignore.case = TRUE)|   
grepl("identify suppliers",sent, ignore.case = TRUE)|   
grepl("disclose supply",sent, ignore.case = TRUE)|   
grepl("disclosing supply",sent, ignore.case = TRUE)|   
grepl(" we operate in",sent, ignore.case = TRUE)|   
grepl(" list of countries",sent, ignore.case = TRUE)|   
grepl("list of regions",sent, ignore.case = TRUE)|   
grepl(" supply chain is based",sent, ignore.case = TRUE)|   
grepl("disclose suppliers",sent, ignore.case = TRUE)|   
grepl("disclosing suppliers",sent, ignore.case = TRUE)|   
grepl("source country of origin of our suppliers",sent, ignore.case = TRUE)|   
grepl("source products from",sent, ignore.case = TRUE)|   
grepl("suppliers based in",sent, ignore.case = TRUE)|   
grepl("source from",sent, ignore.case = TRUE)|   
grepl("diagram below",sent, ignore.case = TRUE)|   
grepl("manufactures in",sent, ignore.case = TRUE)|   
grepl("who are based in ",sent, ignore.case = TRUE)|   
grepl("import from",sent, ignore.case = TRUE)|   
grepl("sourcing hub ",sent, ignore.case = TRUE)|   
grepl("supplier sites based",sent, ignore.case = TRUE)|   
grepl("suppliers sites based",sent, ignore.case = TRUE)|   
grepl("businesses sites based",sent, ignore.case = TRUE)|   
grepl("operations are predominantly located",sent, ignore.case = TRUE)|   
grepl("operations are located in",sent, ignore.case = TRUE)|   
grepl("from manufacturers based in",sent, ignore.case = TRUE)|   
grepl("we source from",sent, ignore.case = TRUE) ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA supply chain disclosure1'=bigvector  
Total$'WFF+MSA supply chain disclosure1' <- as.numeric(as.character(Total$'WFF+MSA supply chain disclosure1'))  
  
##WFF+MSA incidents remediation (revised)  
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl(" if a supplier is found to be",sent, ignore.case = TRUE)|   
grepl("providing remediation direct to the worker affected",sent, ignore.case = TRUE)|   
grepl("backpaymen",sent, ignore.case = TRUE)|   
grepl("support to prosecute",sent, ignore.case = TRUE)|   
grepl("informing senior management",sent, ignore.case = TRUE)|   
grepl(" respond to instances of",sent, ignore.case = TRUE)|   
grepl("instigating corrective action plans",sent, ignore.case = TRUE)|   
grepl("cancelling the contracts of suppliers",sent, ignore.case = TRUE)|   
grepl("supports the supplier to respond",sent, ignore.case = TRUE)|   
grepl("corrective action plan",sent, ignore.case = TRUE)|   
grepl("punitive action",sent, ignore.case = TRUE)|   
grepl("developing a remediation policy",sent, ignore.case = TRUE)|   
grepl("planning to implement a remediation policy",sent, ignore.case = TRUE)|   
grepl("contract termination",sent, ignore.case = TRUE)|   
grepl("termination of the",sent, ignore.case = TRUE)|   
grepl("termination for failure",sent, ignore.case = TRUE)|   
grepl("will not deal with any supplier if",sent, ignore.case = TRUE)|   
grepl("reserves the right to terminate",sent, ignore.case = TRUE)|   
grepl("to terminate",sent, ignore.case = TRUE)|   
grepl("cessation of",sent, ignore.case = TRUE)|   
grepl("cease",sent, ignore.case = TRUE)|   
grepl("curtailment",sent, ignore.case = TRUE)|   
grepl("knowingly involved ",sent, ignore.case = TRUE)|   
grepl("cancel contracts",sent, ignore.case = TRUE)|   
grepl("cancel contract",sent, ignore.case = TRUE)|   
grepl("remedial action",sent, ignore.case = TRUE)|   
grepl("appropriate steps to",sent, ignore.case = TRUE)|   
grepl("continuously improve our processes to fight",sent, ignore.case = TRUE)|   
grepl("we require evidence",sent, ignore.case = TRUE)|   
grepl("action plans",sent, ignore.case = TRUE)  
 ,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA incidents remediation (revised)'=bigvector  
  
bigvector=c()  
  
  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl(" if a supplier is found to be",sent, ignore.case = TRUE)|   
grepl("providing remediation direct to the worker affected",sent, ignore.case = TRUE)|   
grepl("backpaymen",sent, ignore.case = TRUE)|   
grepl("support to prosecute",sent, ignore.case = TRUE)|   
grepl("informing senior management",sent, ignore.case = TRUE)|   
grepl("respond to instances of",sent, ignore.case = TRUE)|   
grepl("instigating corrective action plans",sent, ignore.case = TRUE)|   
grepl("cancelling the contracts of suppliers",sent, ignore.case = TRUE)|   
grepl("supports the supplier to respond",sent, ignore.case = TRUE)|   
grepl("corrective action plan",sent, ignore.case = TRUE)|   
grepl("punitive action",sent, ignore.case = TRUE)|   
grepl("developing a remediation policy",sent, ignore.case = TRUE)|   
grepl("planning to implement a remediation policy",sent, ignore.case = TRUE)|   
grepl("contract termination",sent, ignore.case = TRUE)|   
grepl("termination of the",sent, ignore.case = TRUE)|   
grepl("termination for failure",sent, ignore.case = TRUE)|   
grepl("will not deal with any supplier if",sent, ignore.case = TRUE)|   
grepl("reserves the right to terminate",sent, ignore.case = TRUE)|   
grepl("to terminate",sent, ignore.case = TRUE)|   
grepl("cessation of",sent, ignore.case = TRUE)|   
grepl("cease",sent, ignore.case = TRUE)|   
grepl("curtailment",sent, ignore.case = TRUE)|   
grepl("knowingly involved ",sent, ignore.case = TRUE)|   
grepl("cancel contracts",sent, ignore.case = TRUE)|   
grepl("cancel contract",sent, ignore.case = TRUE)|   
grepl("remedial action",sent, ignore.case = TRUE)|   
grepl("appropriate steps to",sent, ignore.case = TRUE)|   
grepl("continuously improve our processes to fight",sent, ignore.case = TRUE)|   
grepl("we require evidence",sent, ignore.case = TRUE)|   
grepl("action plans",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA incidents remediation (revised)1'=bigvector  
  
Total$`WFF+MSA incidents remediation (revised)1` <- as.numeric(as.character(Total$`WFF+MSA incidents remediation (revised)1`))  
  
  
  
  
## "WFF+MSA whistleblowing mechanism (revised)"  
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("hotline",sent, ignore.case = TRUE)|   
grepl("grievances",sent, ignore.case = TRUE)|   
grepl("suspected incidents",sent, ignore.case = TRUE)|   
grepl("reporting line",sent, ignore.case = TRUE)|   
grepl("anonymous reporting",sent, ignore.case = TRUE)|   
grepl("workers can call",sent, ignore.case = TRUE)|   
grepl("whistleblower",sent, ignore.case = TRUE)|   
grepl("report modern slavery cases",sent, ignore.case = TRUE)|   
grepl("focal point",sent, ignore.case = TRUE)|   
grepl("report any suspic",sent, ignore.case = TRUE)|   
grepl("identify and report",sent, ignore.case = TRUE)|   
grepl("appropriate steps to",sent, ignore.case = TRUE)|  
grepl("we encourage",sent, ignore.case = TRUE)|  
grepl("self-assessment questionnaire",sent, ignore.case = TRUE)|  
grepl("confidential",sent, ignore.case = TRUE)|  
grepl("mechanism for reporting",sent, ignore.case = TRUE)|  
grepl("mechanism to",sent, ignore.case = TRUE)|  
grepl("whistleblowing",sent, ignore.case = TRUE)|  
grepl("whistleblower",sent, ignore.case = TRUE)|   
grepl("whistleblowers",sent, ignore.case = TRUE)|   
grepl("protection",sent, ignore.case = TRUE)  
 ,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA whistleblowing mechanism (revised)'=bigvector  
  
bigvector=c()  
  
  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("hotline",sent, ignore.case = TRUE)|   
grepl("grievances",sent, ignore.case = TRUE)|   
grepl("suspected incidents",sent, ignore.case = TRUE)|   
grepl("reporting line",sent, ignore.case = TRUE)|   
grepl("anonymous reporting",sent, ignore.case = TRUE)|   
grepl("workers can call",sent, ignore.case = TRUE)|   
grepl("whistleblower",sent, ignore.case = TRUE)|   
grepl("report modern slavery cases",sent, ignore.case = TRUE)|   
grepl("focal point",sent, ignore.case = TRUE)|   
grepl("report any suspic",sent, ignore.case = TRUE)|   
grepl("identify and report",sent, ignore.case = TRUE)|   
grepl("appropriate steps to",sent, ignore.case = TRUE)|  
grepl("we encourage",sent, ignore.case = TRUE)|  
grepl("self-assessment questionnaire",sent, ignore.case = TRUE)|  
grepl("confidential",sent, ignore.case = TRUE)|  
grepl("mechanism for reporting",sent, ignore.case = TRUE)|  
grepl("mechanism to",sent, ignore.case = TRUE)|  
grepl("whistleblowing",sent, ignore.case = TRUE)|  
grepl("whistleblower",sent, ignore.case = TRUE)|   
grepl("whistleblowers",sent, ignore.case = TRUE)|   
grepl("protection",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA whistleblowing mechanism (revised)1'=bigvector  
  
Total$`WFF+MSA whistleblowing mechanism (revised)1` <- as.numeric(as.character(Total$`WFF+MSA whistleblowing mechanism (revised)1`))  
  
  
  
  
##"WFF+MSA training (revised)"   
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("provide training",sent, ignore.case = TRUE)|  
grepl("training to",sent, ignore.case = TRUE)|  
grepl("course on modern slavery",sent, ignore.case = TRUE)|  
grepl("training programme",sent, ignore.case = TRUE)|  
grepl("Procurement policy",sent, ignore.case = TRUE)|   
grepl("purchasing policy",sent, ignore.case = TRUE)|   
grepl("purchasing practices",sent, ignore.case = TRUE)|   
grepl("Recruitment policy",sent, ignore.case = TRUE)|   
grepl("Recruitment practices",sent, ignore.case = TRUE)|   
grepl("HR",sent, ignore.case = TRUE)|   
grepl("training provided",sent, ignore.case = TRUE)|   
grepl(" human resources",sent, ignore.case = TRUE)|   
grepl("training provider",sent, ignore.case = TRUE)  
 ,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA training (revised)'=bigvector  
  
bigvector=c()  
  
  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("provide training",sent, ignore.case = TRUE)|  
grepl("training to",sent, ignore.case = TRUE)|  
grepl("course on modern slavery",sent, ignore.case = TRUE)|  
grepl("training programme",sent, ignore.case = TRUE)|  
grepl("Procurement policy",sent, ignore.case = TRUE)|   
grepl("purchasing policy",sent, ignore.case = TRUE)|   
grepl("purchasing practices",sent, ignore.case = TRUE)|   
grepl("Recruitment policy",sent, ignore.case = TRUE)|   
grepl("Recruitment practices",sent, ignore.case = TRUE)|  
grepl("training provided",sent, ignore.case = TRUE)|   
grepl(" human resources",sent, ignore.case = TRUE)|   
grepl("training provider",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA training (revised)1'=bigvector  
  
Total$`WFF+MSA training (revised)1` <- as.numeric(as.character(Total$`WFF+MSA training (revised)1`))  
  
##"WFF+MSA Performance Indicators"   
  
 bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("kpi",sent, ignore.case = TRUE)|   
grepl("kpis",sent, ignore.case = TRUE)|   
grepl("key Performance Indicator",sent, ignore.case = TRUE)|   
grepl("key Performance Indicators",sent, ignore.case = TRUE)|   
grepl("measure impact",sent, ignore.case = TRUE)|   
grepl("action taken",sent, ignore.case = TRUE)|   
grepl("combat modern slavery",sent, ignore.case = TRUE)|   
grepl("measuring the impact of training",sent, ignore.case = TRUE)|   
grepl(" number of audits",sent, ignore.case = TRUE)|   
grepl("on site visits",sent, ignore.case = TRUE)|  
grepl(" suppliers questionnaires",sent, ignore.case = TRUE)|  
grepl("monitoring mechanisms",sent, ignore.case = TRUE)|  
grepl("assess the risk",sent, ignore.case = TRUE)|  
grepl("review our",sent, ignore.case = TRUE)|  
grepl("measure our",sent, ignore.case = TRUE)|  
grepl("review and",sent, ignore.case = TRUE)  
 ,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA Performance Indicators'=bigvector  
  
bigvector=c()  
  
  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("kpi",sent, ignore.case = TRUE)|   
grepl("kpis",sent, ignore.case = TRUE)|   
grepl("key Performance Indicator",sent, ignore.case = TRUE)|   
grepl("key Performance Indicators",sent, ignore.case = TRUE)|   
grepl("measure impact",sent, ignore.case = TRUE)|   
grepl("action taken",sent, ignore.case = TRUE)|   
grepl("combat modern slavery",sent, ignore.case = TRUE)|   
grepl("measuring the impact of training",sent, ignore.case = TRUE)|   
grepl(" number of audits",sent, ignore.case = TRUE)|   
grepl("on site visits",sent, ignore.case = TRUE)|  
grepl(" suppliers questionnaires",sent, ignore.case = TRUE)|  
grepl("monitoring mechanisms",sent, ignore.case = TRUE)|  
grepl("assess the risk",sent, ignore.case = TRUE)|  
grepl("review our",sent, ignore.case = TRUE)|  
grepl("measure our",sent, ignore.case = TRUE)|  
grepl("review and",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA Performance Indicators1'=bigvector  
  
Total$`WFF+MSA Performance Indicators1` <- as.numeric(as.character(Total$`WFF+MSA Performance Indicators1`))  
  
  
##"WFF+MSA Business Performance Indicators"   
 bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("kpi",sent, ignore.case = TRUE)|   
grepl("kpis",sent, ignore.case = TRUE)|   
grepl("Performance Indicator",sent, ignore.case = TRUE)|   
grepl("Performance Indicators",sent, ignore.case = TRUE)|   
grepl("turn-around time",sent, ignore.case = TRUE)|   
grepl("cost of materials",sent, ignore.case = TRUE)|   
grepl("efficiency in production",sent, ignore.case = TRUE)|   
grepl("cheapest goods",sent, ignore.case = TRUE)|   
grepl("cheape goods",sent, ignore.case = TRUE)|   
grepl("sourcing the cheapest goods in the shortest amount of time",sent, ignore.case = TRUE)  
 ,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA Business Performance Indicators'=bigvector  
  
bigvector=c()  
  
  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("kpi",sent, ignore.case = TRUE)|   
grepl("kpis",sent, ignore.case = TRUE)|   
grepl("Performance Indicator",sent, ignore.case = TRUE)|   
grepl("Performance Indicators",sent, ignore.case = TRUE)|   
grepl("turn-around time",sent, ignore.case = TRUE)|   
grepl("cost of materials",sent, ignore.case = TRUE)|   
grepl("efficiency in production",sent, ignore.case = TRUE)|   
grepl("cheapest goods",sent, ignore.case = TRUE)|   
grepl("cheape goods",sent, ignore.case = TRUE)|   
grepl("sourcing the cheapest goods in the shortest amount of time",sent, ignore.case = TRUE)  
 ,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA Business Performance Indicators1'=bigvector  
  
Total$`WFF+MSA Business Performance Indicators1` <- as.numeric(as.character(Total$`WFF+MSA Business Performance Indicators1`))  
  
  
  
  
##"WFF+MSA incidents identified"   
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("incident identified",sent, ignore.case = TRUE)|  
grepl("incident was identified",sent, ignore.case = TRUE)|  
grepl("incidents wereidentified",sent, ignore.case = TRUE)|  
grepl("issues found",sent, ignore.case = TRUE)|  
grepl("issues were identified",sent, ignore.case = TRUE)|  
grepl("issue was identified",sent, ignore.case = TRUE)|  
grepl("found isssue",sent, ignore.case = TRUE)|   
grepl("found incident",sent, ignore.case = TRUE)|   
grepl("issue found",sent, ignore.case = TRUE)|   
grepl("issues found ",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA incidents identified'=bigvector  
  
  
bigvector=c()  
  
  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("incident identified",sent, ignore.case = TRUE)|  
grepl("incident was identified",sent, ignore.case = TRUE)|  
grepl("incidents wereidentified",sent, ignore.case = TRUE)|  
grepl("issues found",sent, ignore.case = TRUE)|  
grepl("issues were identified",sent, ignore.case = TRUE)|  
grepl("issue was identified",sent, ignore.case = TRUE)|  
grepl("found isssue",sent, ignore.case = TRUE)|   
grepl("found incident",sent, ignore.case = TRUE)|   
grepl("issue found",sent, ignore.case = TRUE)|   
grepl("issues found ",sent, ignore.case = TRUE)  
,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA incidents identified1'=bigvector  
  
Total$`WFF+MSA incidents identified1` <- as.numeric(as.character(Total$`WFF+MSA incidents identified1`))  
  
  
  
##"WFF+MSA Identification of risks"  
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("identify risk",sent, ignore.case = TRUE)|  
grepl("risk profile",sent, ignore.case = TRUE)|  
grepl(" risks were identified",sent, ignore.case = TRUE)|  
grepl("risk was identified",sent, ignore.case = TRUE)|  
grepl("found risk",sent, ignore.case = TRUE)|   
grepl("have identified risk",sent, ignore.case = TRUE)|   
grepl("we have identified risk",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA Identification of risks'=bigvector  
  
  
bigvector=c()  
  
  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("identify risk",sent, ignore.case = TRUE)|  
grepl("risk profile",sent, ignore.case = TRUE)|  
grepl(" risks were identified",sent, ignore.case = TRUE)|  
grepl("risk was identified",sent, ignore.case = TRUE)|  
grepl("found risk",sent, ignore.case = TRUE)|   
grepl("have identified risk",sent, ignore.case = TRUE)|   
grepl("we have identified risk",sent, ignore.case = TRUE)  
,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA Identification of risks1'=bigvector  
  
Total$`WFF+MSA Identification of risks1` <- as.numeric(as.character(Total$`WFF+MSA Identification of risks1`))  
  
##"WFF+MSA risk management (revised)"  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("due diligence",sent, ignore.case = TRUE)|  
grepl("planning to implement",sent, ignore.case = TRUE)|  
grepl("continuous improvement programs",sent, ignore.case = TRUE)|  
grepl("audit of suppliers",sent, ignore.case = TRUE)|  
grepl("continuously engaging with suppliers",sent, ignore.case = TRUE)|   
grepl("on-site visits",sent, ignore.case = TRUE)|   
grepl("audits of suppliers",sent, ignore.case = TRUE)|   
grepl("audits",sent, ignore.case = TRUE)|   
grepl("audit",sent, ignore.case = TRUE)|   
grepl("monitor",sent, ignore.case = TRUE)|   
grepl("third party",sent, ignore.case = TRUE)|   
grepl("verif",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA risk management (revised)'=bigvector  
  
  
bigvector=c()  
  
  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("due diligence",sent, ignore.case = TRUE)|  
grepl("planning to implement",sent, ignore.case = TRUE)|  
grepl("continuous improvement programs",sent, ignore.case = TRUE)|  
grepl("audit of suppliers",sent, ignore.case = TRUE)|  
grepl("continuously engaging with suppliers",sent, ignore.case = TRUE)|   
grepl("on-site visits",sent, ignore.case = TRUE)|   
grepl("audits of suppliers",sent, ignore.case = TRUE)|   
grepl("audits",sent, ignore.case = TRUE)|   
grepl("audit",sent, ignore.case = TRUE)|   
grepl("monitor",sent, ignore.case = TRUE)|   
grepl("third party",sent, ignore.case = TRUE)|   
grepl("verif",sent, ignore.case = TRUE)  
,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA risk management (revised)1'=bigvector  
  
Total$`WFF+MSA risk management (revised)1` <- as.numeric(as.character(Total$`WFF+MSA risk management (revised)1`))  
  
## "WFF+MSA risk assessment"   
  
bigvector=c()  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("assess risk",sent, ignore.case = TRUE)|  
grepl("risk-based questionnaires",sent, ignore.case = TRUE)|  
grepl("risk management database",sent, ignore.case = TRUE)|  
grepl("risk management tool",sent, ignore.case = TRUE)|  
grepl("Maplecrofts",sent, ignore.case = TRUE)|   
grepl("Sedex",sent, ignore.case = TRUE)|   
grepl("risk matrix",sent, ignore.case = TRUE)|   
grepl("conducting research",sent, ignore.case = TRUE)|   
grepl("risk assessment",sent, ignore.case = TRUE)|   
grepl("assessing",sent, ignore.case = TRUE)  
,vector<-sent,x<-0)};   
{ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA risk assessment'=bigvector  
  
  
bigvector=c()  
  
  
for (i in 1:length(Total$text)) { doc = Total$text[i]; sents = tokenize\_sentences(doc); vector=c(); for (sent in sents[[1]]) {ifelse(  
grepl("assess risk",sent, ignore.case = TRUE)|  
grepl("risk-based questionnaires",sent, ignore.case = TRUE)|  
grepl("risk management database",sent, ignore.case = TRUE)|  
grepl("risk management tool",sent, ignore.case = TRUE)|  
grepl("Maplecrofts",sent, ignore.case = TRUE)|   
grepl("Sedex",sent, ignore.case = TRUE)|   
grepl("risk matrix",sent, ignore.case = TRUE)|   
grepl("conducting research",sent, ignore.case = TRUE)|   
grepl("risk assessment",sent, ignore.case = TRUE)|   
grepl("assessing",sent, ignore.case = TRUE)  
,vector<-1,x<-0)}; {ifelse(is.not.null(vector[1]),{print("length(bigvector)");print(length(bigvector));print("i"); print(i); bigvector[i] <- vector[1]},bigvector[i] <- "")}}   
Total$'WFF+MSA risk assessment1'=bigvector  
  
Total$`WFF+MSA risk assessment1` <- as.numeric(as.character(total$`WFF+MSA risk assessment1`))  
 Statement Homepage Link` <- as.numeric(as.character(Only\_clean\_reports$`BHRRC+MSA Statement Homepage Link`))

**"BHRRC+MSS"**

Only\_clean\_reports$'BHRRC+MSS'<- 1  
  
Only\_clean\_reports$`BHRRC+MSS` <- as.numeric(as.character(Only\_clean\_reports$`BHRRC+MSS`))

**FEB DATA**

**Text analysis on metric and values:**

comment<-as.factor(comment)

metric<-as.factor(metric)

metric<-UK\_Modern\_Slavery\_Act\_Research$`METRIC NAME`

comment<-UK\_Modern\_Slavery\_Act\_Research$COMMENTS

options(stringsAsFactors = F)

data = data.frame(

metric = metric,

comment = comment)

custom.stopwords = c(stopwords("english"), "http","wikirate.org","utc", "modern",

"slaveri")

data$comment <- removeNumbers(data$comment)

data$comment <- content\_transformer(tolower)(data$comment)

data$comment <- removeWords((data$comment), stopwords("english"))

data$comment<- removePunctuation(data$comment)

data$comment <- stripWhitespace(data$comment)

data$comment <- stemDocument((data$comment), language = "english")

library(tidytext)

tidy\_data <- data %>%

unnest\_tokens(word, comment) %>%

group\_by(word) %>%

filter(n() > 10) %>%

ungroup()

tidy\_data

tidy\_data<- tidy\_data %>%

anti\_join(stop\_words)

tidy\_data<- tidy\_data %>%

tidy\_data %>%

count(metric, word, sort = TRUE) %>%

anti\_join(get\_stopwords()) %>%

group\_by(metric) %>%

top\_n(20) %>%

ungroup()

tidy\_data %>%

count(word, sort = TRUE)

tidy\_data %>%

count(metric, word, sort = TRUE) %>%

anti\_join(get\_stopwords()) %>%

group\_by(metric) %>%

top\_n(20) %>%

ungroup() %>%

ggplot(aes(reorder\_within(word, n, metric), n,

fill = metric

))+

geom\_col(alpha = 0.8, show.legend = FALSE) +

scale\_x\_reordered() +

coord\_flip() +

facet\_wrap(~metric, scales = "free") +

scale\_y\_continuous(expand = c(0, 0)) +

labs(

x = NULL, y = "Word count",

title = "Most frequent words after removing stop words",

subtitle = "Words like 'said' occupy similar ranks but other words are quite different"

)

**Merged value comment**

comment<-as.factor(comment)

metric<-UK\_Modern\_Slavery\_Act\_Research$`METRIC NAME`

metric<-as.factor(metric)

comment<-UK\_Modern\_Slavery\_Act\_Research$COMMENTS

value<-UK\_Modern\_Slavery\_Act\_Research$VALUE

value<-as.factor(value)

options(stringsAsFactors = F)

data1 = data.frame(

metric = metric,

comment = comment,

value=value)

custom.stopwords = c(stopwords("english"), "http","wikirate.org","utc", "modern",

"slaveri","slavery")

data1$comment <- removeNumbers(data1$comment)

data1$comment <- content\_transformer(tolower)(data1$comment)

data1$comment <- removeWords((data1$comment), stopwords("english"))

data1$comment<- removePunctuation(data1$comment)

data1$comment <- stripWhitespace(data1$comment)

data1$comment <- stemDocument((data1$comment), language = "english")

library(tidytext)

tidy\_data2 <- data1 %>%

unnest\_tokens(word, comment,) %>%

group\_by(value) %>%

filter(n() > 10) %>%

ungroup()

tidy\_data2

tidy\_data2<- tidy\_data2 %>%

anti\_join(stop\_words)

tidy\_data2 %>%

count(metric, value, word, sort = TRUE) %>%

anti\_join(get\_stopwords()) %>%

group\_by(metric) %>%

top\_n(20) %>%

ungroup()

tidy\_data2 %>%

count(metric, value, word, sort = TRUE) %>%

filter(metric %in% "WFF+MSA whistleblowing mechanism (revised)") %>%

anti\_join(get\_stopwords()) %>%

group\_by(value) %>%

top\_n(30) %>%

ungroup() %>%

ggplot(aes(reorder\_within(word, n, value), n,

fill = value

))+

geom\_col(alpha = 0.8, show.legend = FALSE) +

scale\_x\_reordered() +

coord\_flip() +

facet\_wrap(~value, scales = "free") +

scale\_y\_continuous(expand = c(0, 0)) +

labs(

x = NULL, y = "Word count",

title = "Most frequent words acrosss each value of the metric ",

subtitle = "WFF+MSA whistleblowing mechanism (revised)"

)

**Predict next word function based on the Only\_clean reports words**

clean <- function(x){

x <- iconv(x, "latin1", "UTF-8")

x <- gsub("a\u0080\u0099", "'", x, fixed=TRUE)

x <- gsub("a\u0080\u0093", " ", x, fixed=TRUE)

x <- gsub("a\u0080\u0098", " ", x, fixed=TRUE)

x <- gsub("a\u0080\u009c", " ", x, fixed=TRUE)

x <- gsub("a\u0080\u009d", " ", x, fixed=TRUE)

x <- gsub("a\u0080\u0094", " ", x, fixed=TRUE)

x <- gsub("a\u0080", " ", x, fixed=TRUE)

x <- gsub("<", " ", x)

x <- gsub(">", " ", x)

x <- gsub("\\. |\\.$", " <EOS> ", x)

x <- gsub("\\? |\\?$", " <EOS> ", x)

x <- gsub("\\! |\\!$", " <EOS> ", x)

x <- gsub("?", " ", fixed = TRUE, x)

x <- gsub("???Ts", " ", fixed = TRUE, x)

x <- gsub(" [b-hj-z] ", " ", x)

x <- gsub(" [B-HJ-Z] ", " ", x)

x <- gsub("[^[:alnum:][:space:]'<>]", " ", x)

x <- gsub("^ \*'| +'|' +", " ", x) # remove apostrophes except the apostrophes in the contraction words

return(x)

}

Only\_clean\_reports$text1<-clean(Only\_clean\_reports$text)

profanity <- as.character(read.csv("C:/Users/Utilisateur/Desktop/full-list-of-bad-words-banned-by-google1.csv", header = FALSE)$V1)

### N-gram Model

The entire corpus was tokenized into unigrams, bigrams and trigrams.

```{r}

library(quanteda)

# Writie a function to tokenize the corpus into N-grams.

ToTokenize <- function(object, n){

tokensAll <- tokens(object, remove\_numbers = TRUE,

remove\_symbols = TRUE, remove\_separators = TRUE,

remove\_twitter = FALSE, remove\_hyphens = TRUE, remove\_url = TRUE)

NoBadWord <- tokens\_select(tokensAll, c(profanity), selection = "remove", case\_insensitive = TRUE)

ng <- tokens\_ngrams(NoBadWord, n, concatenator = " ")

newDfm <- dfm(ng)

newDfm <- dfm\_select(newDfm, "^[e][o][s]|[e][o][s]$| [e][o][s] ", selection="remove", valuetype = "regex")

return(newDfm)

}

dfm1 <- ToTokenize(theSentences, 1)

saveRDS(dfm1, "dfm1.rds")

rm(dfm1)

dfm2 <- ToTokenize(theSentences, 2)

saveRDS(dfm2, "dfm2.rds")

rm(dfm2)

dfm3 <- ToTokenize(theSentences, 3)

saveRDS(dfm3, "dfm3.rds")

rm(dfm3)

dfm4 <- ToTokenize(theSentences, 4)

saveRDS(dfm4, "dfm4.rds")

rm(dfm4)

dfm5 <- ToTokenize(theSentences, 5)

saveRDS(dfm5, "dfm5.rds")

rm(dfm5)

rm(theSentences)

# Convert dfm to data table.

ToDT <- function(object, n){

df <- data.frame(feature = featnames(object), frequency = colSums(object),

row.names = NULL, stringsAsFactors = FALSE)

df$base <- word(string = df$feature, start = 1, end = n-1, sep = fixed(" "))

df$predict <- word(string = df$feature, start = n, end = n, sep = fixed(" "))

DT <- as.data.table(df)

DT <- DT[, c("feature") := NULL][order(-frequency)]

return(DT)

}

n1 <- readRDS("dfm1.rds")

df1 <- data.frame(base = featnames(n1), frequency = colSums(n1),

row.names = NULL, stringsAsFactors = FALSE)

DT1 <- as.data.table(df1)[order(-frequency)]

saveRDS(DT1, "DT1.rds")

rm(n1); rm(df1); rm(DT1)

n2 <- readRDS("dfm2.rds")

DT2 <- ToDT(n2, 2)

saveRDS(DT2, "DT2.rds")

rm(n2); rm(DT2)

n3 <- readRDS("dfm3.rds")

DT3 <- ToDT(n3, 3)

saveRDS(DT3, "DT3.rds")

rm(n3); rm(DT3)

n4 <- readRDS("dfm4.rds")

DT4 <- ToDT(n4, 4)

saveRDS(DT4, "DT4.rds")

rm(n4); rm(DT4)

n5 <- readRDS("dfm5.rds")

DT5 <- ToDT(n5, 5)

saveRDS(DT5, "DT5.rds")

rm(n5); rm(DT5)

```

```{r}

setwd("C:/Users/Utilisateur/Desktop/Thesis data R")

DT1 <- readRDS("DT1.rds")

DT2 <- readRDS("DT2.rds")

DT3 <- readRDS("DT3.rds")

DT4 <- readRDS("DT4.rds")

DT5 <- readRDS("DT5.rds")

DT2 <- DT2[frequency != 1]

DT3 <- DT3[frequency != 1]

DT4 <- DT4[frequency != 1]

DT5 <- DT5[frequency != 1]

saveRDS(DT2, "C:/Users/Utilisateur/Desktop/Thesis data R/no singletons/DT2.rds")

saveRDS(DT3, "C:/Users/Utilisateur/Desktop/Thesis data R/no singletons/DT3.rds")

saveRDS(DT4, "C:/Users/Utilisateur/Desktop/Thesis data R/no singletons/DT4.rds")

saveRDS(DT5, "C:/Users/Utilisateur/Desktop/Thesis data R/no singletons/DT5.rds")

```

```{r}

PredictNext <- function(input){

input <- tolower(input)

input <- unlist(strsplit(as.character(input), ' '))

n <- length(input)

if(n >= 4 & nrow(DT5[base == paste(input[n-3], input[n-2], input[n-1], input[n], sep = " "),]) > 0){

new <- DT5[.(paste(input[n-3], input[n-2], input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT4[base == paste(input[n-2], input[n-1], input[n], sep = " "),]) > 0) {

new <- DT4[.(paste(input[n-2], input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT3[base == paste(input[n-1], input[n], sep = " "),]) > 0){

new <- DT3[.(paste(input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT2[base == paste(input[n], sep = ""),]) > 0){

new <- DT2[.(paste(input[n], sep = "")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(n == 3 & nrow(DT4[base == paste(input[n-2], input[n-1], input[n], sep = " "),]) > 0){

new <- DT4[.(paste(input[n-2], input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT3[base == paste(input[n-1], input[n], sep = " "),]) > 0) {

new <- DT3[.(paste(input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT2[base == paste(input[n], sep = ""),]) > 0){

new <- DT2[.(paste(input[n], sep = "")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(n == 2 & nrow(DT3[base == paste(input[n-1], input[n], sep = " "),]) > 0){

new <- DT3[.(paste(input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT2[base == paste(input[n], sep = ""),]) > 0) {

new <- DT2[.(paste(input[n], sep = "")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(n == 1 & nrow(DT2[base == paste(input[n], sep = " "),]) > 0){

new <- DT2[.(paste(input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else{

return("Unknown")

}

}

PredictNext("list")

library(tm)

dfm1 <- ToTokenize(Only\_clean\_reports$text1, 1)

saveRDS(dfm1, "dfm1.rds")

rm(dfm1)

dfm2 <- ToTokenize(Only\_clean\_reports$text1, 2)

saveRDS(dfm2, "dfm2.rds")

rm(dfm2)

dfm3 <- ToTokenize(Only\_clean\_reports$text1, 3)

saveRDS(dfm3, "dfm3.rds")

rm(dfm3)

dfm4 <- ToTokenize(Only\_clean\_reports$text1, 4)

saveRDS(dfm4, "dfm4.rds")

rm(dfm4)

dfm5 <- ToTokenize(Only\_clean\_reports$text1, 5)

saveRDS(dfm5, "dfm5.rds")

rm(dfm5)

# Convert dfm to data table.

ToDT <- function(object, n){

df <- data.frame(feature = featnames(object), frequency = colSums(object),

row.names = NULL, stringsAsFactors = FALSE)

df$base <- word(string = df$feature, start = 1, end = n-1, sep = fixed(" "))

df$predict <- word(string = df$feature, start = n, end = n, sep = fixed(" "))

DT <- as.data.table(df)

DT <- DT[, c("feature") := NULL][order(-frequency)]

return(DT)

}

n1 <- readRDS("dfm1.rds")

df1 <- data.frame(base = featnames(n1), frequency = colSums(n1),

row.names = NULL, stringsAsFactors = FALSE)

DT1 <- as.data.table(df1)[order(-frequency)]

saveRDS(DT1, "DT1.rds")

rm(n1); rm(df1); rm(DT1)

n2 <- readRDS("dfm2.rds")

DT2 <- ToDT(n2, 2)

saveRDS(DT2, "DT2.rds")

rm(n2); rm(DT2)

n3 <- readRDS("dfm3.rds")

DT3 <- ToDT(n3, 3)

saveRDS(DT3, "DT3.rds")

rm(n3); rm(DT3)

n4 <- readRDS("dfm4.rds")

DT4 <- ToDT(n4, 4)

saveRDS(DT4, "DT4.rds")

rm(n4); rm(DT4)

n5 <- readRDS("dfm5.rds")

DT5 <- ToDT(n5, 5)

saveRDS(DT5, "DT5.rds")

rm(n5); rm(DT5)

```

```{r}

setwd("C:/Users/Utilisateur/Desktop/Thesis data R")

DT1 <- readRDS("DT1.rds")

DT2 <- readRDS("DT2.rds")

DT3 <- readRDS("DT3.rds")

DT4 <- readRDS("DT4.rds")

DT5 <- readRDS("DT5.rds")

DT2 <- DT2[frequency != 1]

DT3 <- DT3[frequency != 1]

DT4 <- DT4[frequency != 1]

DT5 <- DT5[frequency != 1]

saveRDS(DT2, "C:/Users/Utilisateur/Desktop/Thesis data R/no singletons/DT2.rds")

saveRDS(DT3, "C:/Users/Utilisateur/Desktop/Thesis data R/no singletons/DT3.rds")

saveRDS(DT4, "C:/Users/Utilisateur/Desktop/Thesis data R/no singletons/DT4.rds")

saveRDS(DT5, "C:/Users/Utilisateur/Desktop/Thesis data R/no singletons/DT5.rds")

```

```{r}

PredictNext <- function(input){

input <- tolower(input)

input <- unlist(strsplit(as.character(input), ' '))

n <- length(input)

if(n >= 4 & nrow(DT5[base == paste(input[n-3], input[n-2], input[n-1], input[n], sep = " "),]) > 0){

new <- DT5[.(paste(input[n-3], input[n-2], input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT4[base == paste(input[n-2], input[n-1], input[n], sep = " "),]) > 0) {

new <- DT4[.(paste(input[n-2], input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT3[base == paste(input[n-1], input[n], sep = " "),]) > 0){

new <- DT3[.(paste(input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT2[base == paste(input[n], sep = ""),]) > 0){

new <- DT2[.(paste(input[n], sep = "")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(n == 3 & nrow(DT4[base == paste(input[n-2], input[n-1], input[n], sep = " "),]) > 0){

new <- DT4[.(paste(input[n-2], input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT3[base == paste(input[n-1], input[n], sep = " "),]) > 0) {

new <- DT3[.(paste(input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT2[base == paste(input[n], sep = ""),]) > 0){

new <- DT2[.(paste(input[n], sep = "")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(n == 2 & nrow(DT3[base == paste(input[n-1], input[n], sep = " "),]) > 0){

new <- DT3[.(paste(input[n-1], input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(nrow(DT2[base == paste(input[n], sep = ""),]) > 0) {

new <- DT2[.(paste(input[n], sep = "")), head(.SD, 3), on = "base"]

return(new[, predict])

} else if(n == 1 & nrow(DT2[base == paste(input[n], sep = " "),]) > 0){

new <- DT2[.(paste(input[n], sep = " ")), head(.SD, 3), on = "base"]

return(new[, predict])

} else{

return("Unknown")

}

}

PredictNext("list")

**Machine learning algorithm : METRIC NAME AND COMMENTS**

library(gdata)

library(wordcloud)

library(tm)

library(quanteda)

library(syuzhet)

library(SnowballC)

library(tidytext)

library(devtools)

library(caret)

library(text2vec)

library(glmnet)

library(e1071)

library(klaR)

library(pROC)

library(data.table)

library(magrittr)

library(qdap)

library(readr)

library(qdapTools)

library(qdapRegex)

library(randomcoloR)

library(class)

**### First train and test on the Wikirate \_ UK\_Modern\_Slavery\_Act\_Research data**

library(readxl)

options(stringsAsFactors = F)

docs<-UK\_Modern\_Slavery\_Act\_Research

docs$type<- as.factor(docs$`METRIC NAME`)

is.na(docs$type) <- docs$type == "type"

docs$type <- factor(docs$type)

levels(docs$type)

plot(docs$type, col = c("blue", "blue", "blue", "blue", "red", "red", "red", "red", "green", "green", "green", "green", "yellow", "yellow", "yellow", "yellow"))

options(stringsAsFactors = FALSE)

df<-UK\_Modern\_Slavery\_Act\_Research

df$posts<-df$COMMENTS

df$type<-df$`METRIC NAME`

docs1 <- Corpus(VectorSource(df$posts))

custom.stopwords = c(stopwords("english"), "i","like","just","can","im","can","the","one","get","ive","just","know")

docs1 <- tm\_map(docs1, content\_transformer(tolower))

docs1 <- tm\_map(docs1, removeNumbers)

docs1 <- tm\_map(docs1, removeWords, stopwords("english"))

docs1 <- tm\_map(docs1, removePunctuation)

docs1 <- tm\_map(docs1, stripWhitespace)

docs1 <- tm\_map(docs1, stemDocument, language = "english")

dtm <- DocumentTermMatrix(docs1)

mat.df <- as.data.frame(data.matrix(dtm), stringsAsfactors = FALSE)

mat.df <- cbind(mat.df, df$type, row.names = NULL)

colnames(mat.df)[ncol(mat.df)] <- "type"

train <- sample(nrow(mat.df), ceiling(nrow(mat.df) \* .70))

test <- (1:nrow(mat.df))[- train]

cl <- mat.df[,"type"]

modeldata <- mat.df[,!colnames(mat.df) %in% "type"]

knn.pred <- knn(modeldata[train, ], modeldata[test, ], cl[train])

conf.mat <- table("Predictions" = knn.pred, Actual = cl[test])

conf.mat

(accuracy <- sum(diag(conf.mat))/length(test) \* 100)

**### Train on the on the WIKIRATEdata set and test on the new created data sets with metrics**

options(stringsAsFactors = F)

docs2<-Total1cl

docs2$type2<- as.factor(docs2$variable)

is.na(docs2$type2) <- docs2$type2 == "type2"

docs2$type2 <- factor(docs2$type2)

levels(docs2$type2)

plot(docs2$type2, col = c("blue", "blue", "blue", "blue", "red", "red", "red", "red", "green", "green", "green", "green", "yellow", "yellow", "yellow", "yellow"))

options(stringsAsFactors = FALSE)

df2<-Total1cl

df2$posts2<-df2$value

df2$type2<-df2$variable

docs3 <- Corpus(VectorSource(df2$posts2))

custom.stopwords = c(stopwords("english"), "i","like","just","can","im","can","the","one","get","ive","just","know")

docs3 <- tm\_map(docs1, content\_transformer(tolower))

docs3 <- tm\_map(docs1, removeNumbers)

docs3 <- tm\_map(docs1, removeWords, stopwords("english"))

docs3 <- tm\_map(docs1, removePunctuation)

docs3 <- tm\_map(docs1, stripWhitespace)

docs3 <- tm\_map(docs1, stemDocument, language = "english")

dtm2 <- DocumentTermMatrix(docs3)

mat.df2 <- as.data.frame(data.matrix(dtm2), stringsAsfactors = FALSE)

mat.df2 <- cbind(mat.df2, df2$type2, row.names = NULL)

colnames(mat.df2)[ncol(mat.df2)] <- "type2"

train <- sample(nrow(mat.df), ceiling(nrow(mat.df) \* 1)) ##this comes form the wiki data

test2 <- sample(nrow(mat.df2), ceiling(nrow(mat.df2) \* 0.5))

cl2 <- mat.df2[,"type2"]

modeldata2 <- mat.df2[,!colnames(mat.df2) %in% "type2"]

knn.pred2 <- knn(modeldata1[train, ], modeldata1[test2, ], cl1[train])

conf.mat2 <- table("Predictions" = knn.pred2, Actual = cl1[test2])

conf.mat2

(accuracy2 <- sum(diag(conf.mat2))/length(test2) \* 100)

Extract from the results:

