Algorithm Aversion

Kopkow, Angermaier, Rohrer, Sonnleitner

2/17/2022

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#summary(cars)
```

Including Plots

You can also embed plots, for example:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

Algortihm Aversion

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidytext)
library(textdata)
library(dplyr)
library(vader)
library(academictwitteR)
library(data.table)
```

```
##
## Attache Paket: ’data.table’
```

```
## Die folgenden Objekte sind maskiert von ’package:dplyr’:
##
##     between, first, last
```

```
## Das folgende Objekt ist maskiert ’package:purrr’:
##
##     transpose
```

```
library(readr)
```

```
## New names:
## * ‘‘ -> ...1
```

```
## Rows: 143271 Columns: 16
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (3): text, VADERclass, topic
## dbl  (12): ...1, index, id, VADER, Business, Social.Media, Technology, Immut...
## date  (1): created_at
```

```
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
## ‘summarise()‘ has grouped output by ’alg.data$topic’, ’alg.data$Year’. You can override using the ‘.g
```

```
## ‘summarise()‘ has grouped output by ’topic’, ’Year’. You can override using the ‘.groups‘ argument.
```

```
## ‘summarise()‘ has grouped output by ’Year’. You can override using the ‘.groups‘ argument.
```

```
listofdfs <- list()

#data group for cummulated dataframe

for (i in c(7:13)){
  alg.data %>%
    filter(alg.data[,i]>=1)%>%
    mutate(Topic = colnames(alg.data)[i], VADERclass=as.factor(VADERclass))%>%
    group_by(Year,Topic,VADERclass, .drop=FALSE)%>%
    summarise(Sent = n(), .groups = "drop") ->listofdfs[[i]]
}
summ.cumm <- bind_rows(listofdfs)


#adds percentage to cummulated DF
summ.cumm <-
  summ.cumm %>%
  group_by(Topic, Year) %>%
  mutate(All = sum(Sent),percent=(100*Sent/All))

summ.cumm
```

```
## # A tibble: 324 x 6
## # Groups:   Topic, Year [84]
##     Year Topic    VADERclass  Sent   All percent
##    <dbl> <chr>    <fct>      <int> <int>   <dbl>
##  1  2010 Business Aversive       2   297   0.673
##  2  2010 Business Negative      14   297   4.71
##  3  2010 Business Neutral      226   297  76.1
##  4  2010 Business Positive      55   297  18.5
##  5  2011 Business Aversive       7   337   2.08
##  6  2011 Business Negative      17   337   5.04
##  7  2011 Business Neutral      217   337  64.4
##  8  2011 Business Positive      96   337  28.5
##  9  2012 Business Aversive       4   357   1.12
## 10  2012 Business Negative      36   357  10.1
## # ... with 314 more rows
```

```
#data group for topic oriented dataframe

listofdfs <- list()

for (i in c(7:13)){
  alg.data %>%
    filter(alg.data[,i]>=1)%>%
    mutate(Topic = colnames(alg.data)[i])%>%
    group_by( VADERclass, Topic) %>%
    summarise(Sent = n()) ->listofdfs[[i]]
}
```

```
## `summarise()` has grouped output by 'VADERclass'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'VADERclass'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'VADERclass'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'VADERclass'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'VADERclass'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'VADERclass'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'VADERclass'. You can override using the `.groups` argument.
```

```
summ.topic <- bind_rows(listofdfs)

#adds percentage to topicwise DF
summ.topic<-
  summ.topic %>%
  group_by(Topic) %>%
  mutate(All = sum(Sent),percent=(100*Sent/All))


summ.topic
```

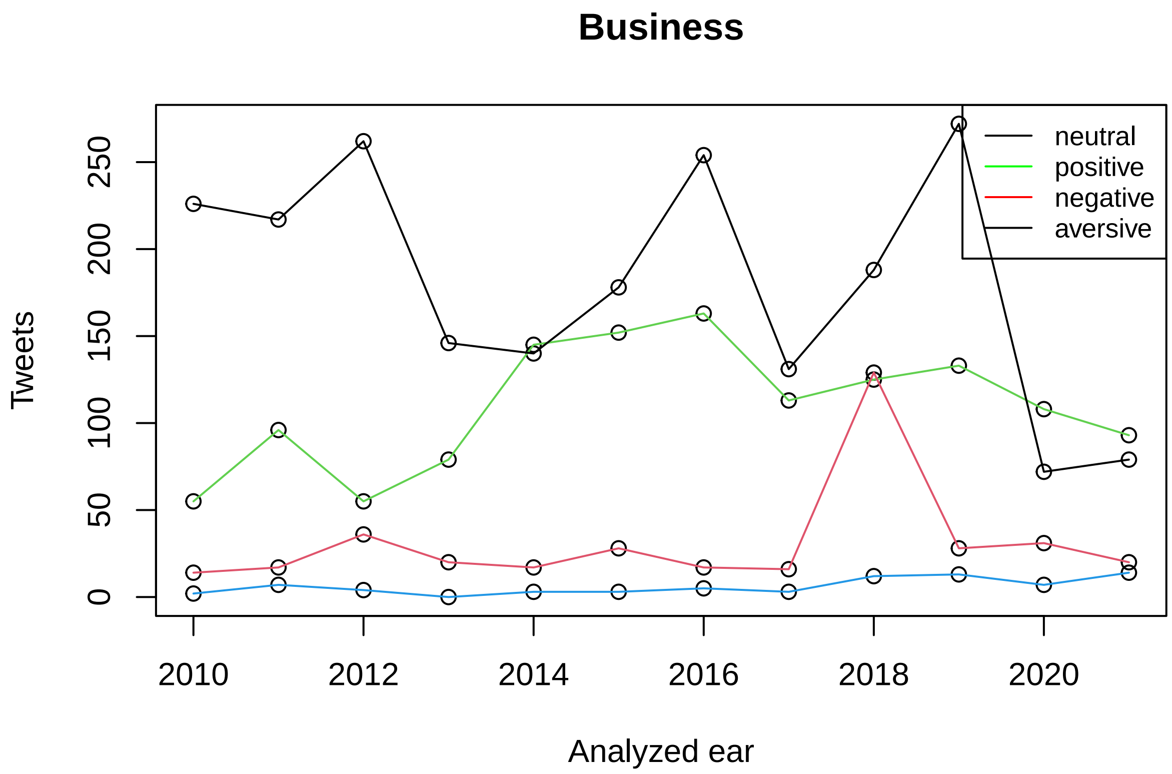
```
## # A tibble: 27 x 5
## # Groups:   Topic [7]
##    VADERclass Topic          Sent   All percent
##    <chr>      <chr>         <int> <int>   <dbl>
##  1 Aversive   Business         73  3928    1.86
##  2 Negative   Business        373  3928    9.50
##  3 Neutral    Business       2165  3928   55.1
##  4 Positive   Business       1317  3928   33.5
##  5 Aversive   Social.Media   1601 43621    3.67
##  6 Negative   Social.Media   5918 43621   13.6
##  7 Neutral    Social.Media  23501 43621   53.9
##  8 Positive   Social.Media  12601 43621   28.9
##  9 Aversive   Technology      130  7730    1.68
## 10 Negative   Technology      475  7730    6.14
## # ... with 17 more rows
```

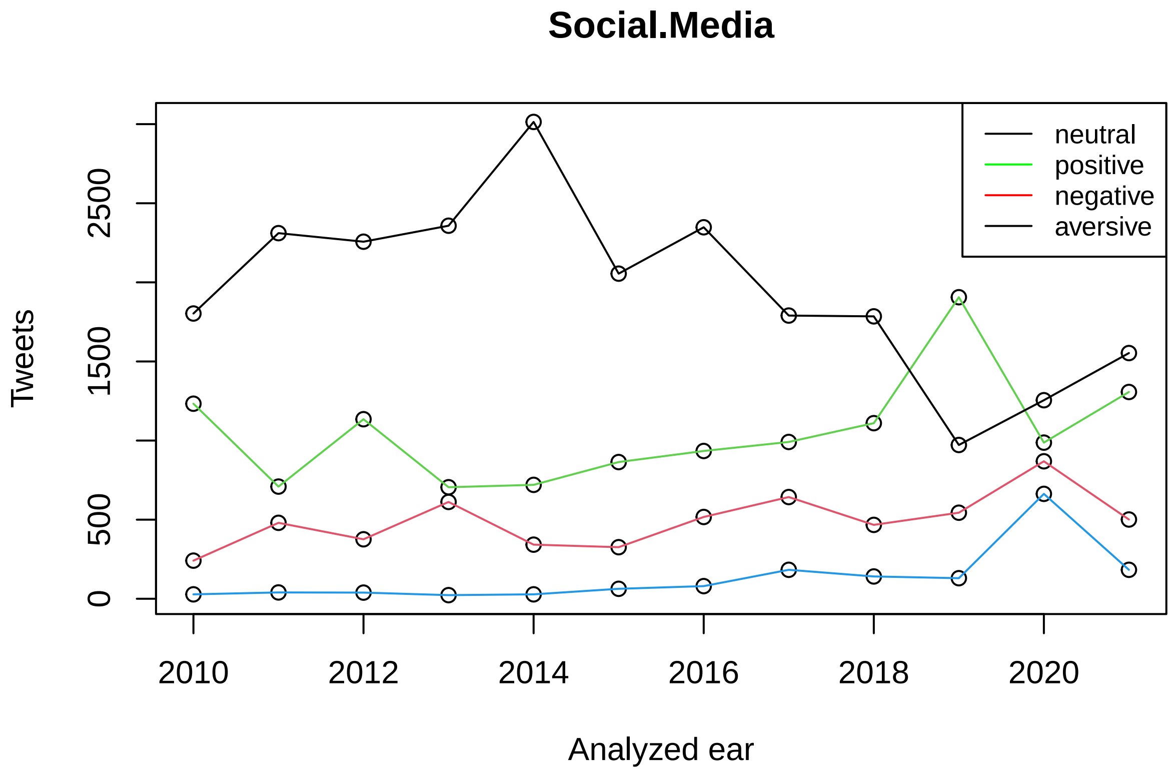
```
#--------------Test wordgroups---------------------
listofdfs <- list()


for (i in c(7:13)){
  alg.data %>%
    filter(alg.data[,i]>=1)%>%
    mutate(Topic = colnames(alg.data)[i])%>%
    sample_n(size = 20)%>%
  group_by( VADERclass, Topic)  ->listofdfs[[i]]
}
Wordgrouptest<- bind_rows(listofdfs)
```

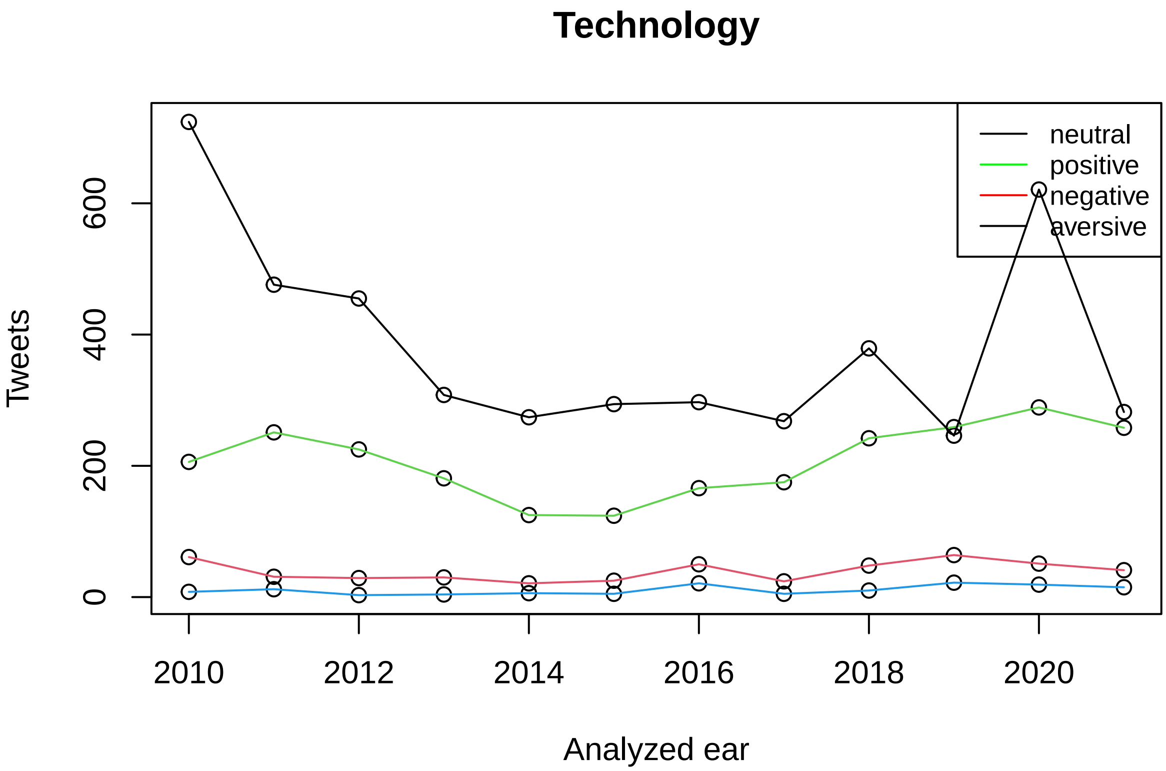
```
#--------------Bootstraping---------------------
listofdfs <- list()


for (i in c(7:13)){
  alg.data %>%
  filter(alg.data[,i]>=1)%>%
  mutate(Topic = colnames(alg.data)[i])%>%
    sample_n(size = 100)%>%
    group_by( VADERclass, Topic) %>%

  summarise(Sent = n()) ->listofdfs[[i]]
}
```

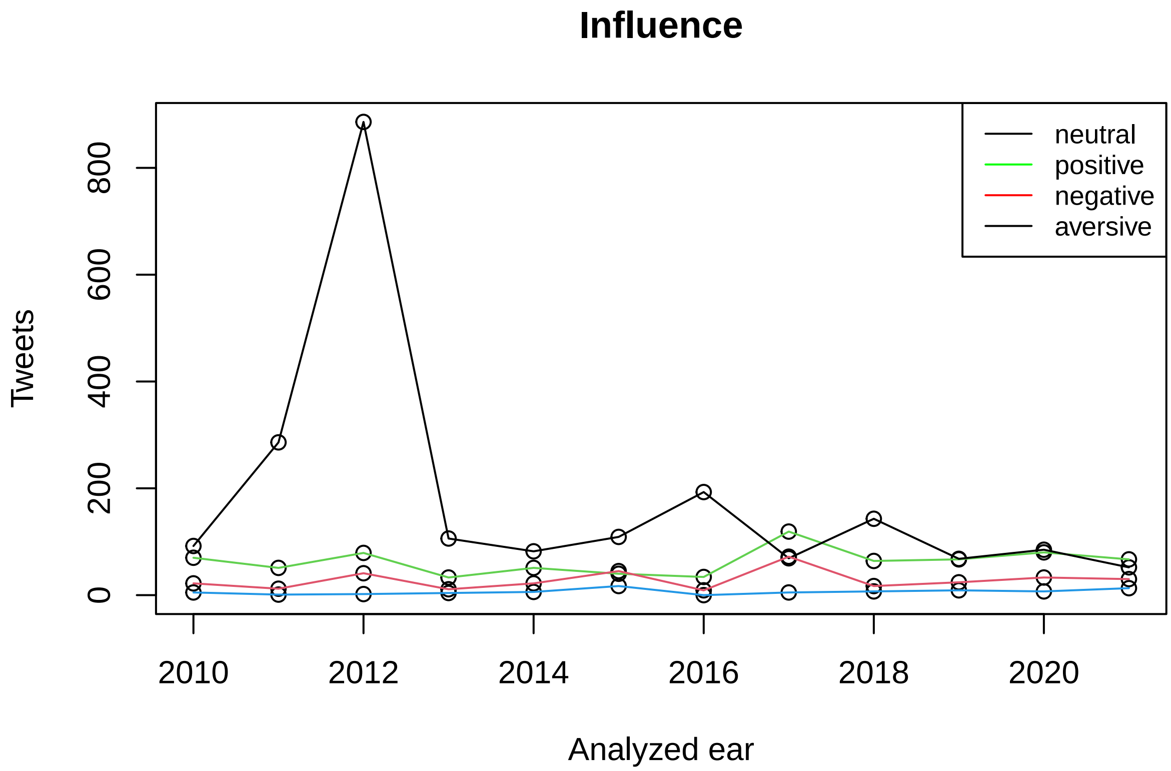
```
## ‘summarise()‘ has grouped output by ’VADERclass’. You can override using the ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’VADERclass’. You can override using the ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’VADERclass’. You can override using the ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’VADERclass’. You can override using the ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’VADERclass’. You can override using the ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’VADERclass’. You can override using the ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’VADERclass’. You can override using the ‘.groups‘ argument.
```

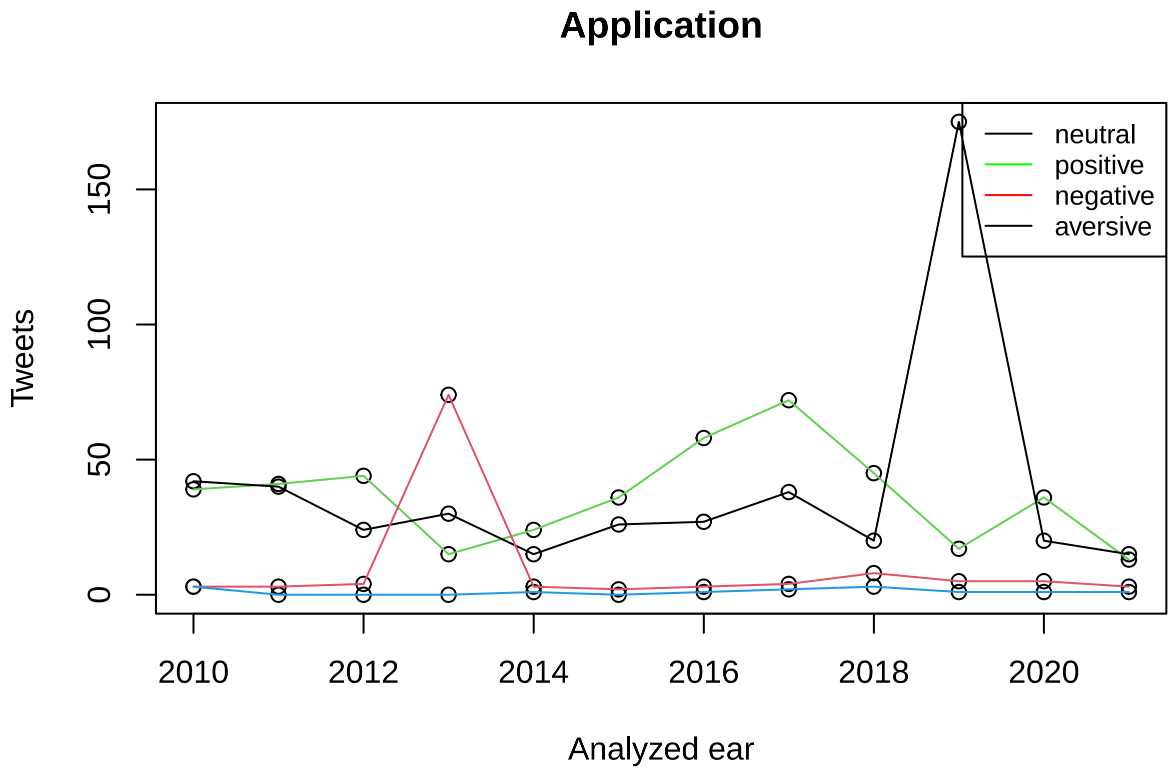
```
bootstrap<- bind_rows(listofdfs)

bootstrap<-
  bootstrap %>%
  group_by(Topic) %>%
  mutate(All = sum(Sent),percent=(100*Sent/All))
```

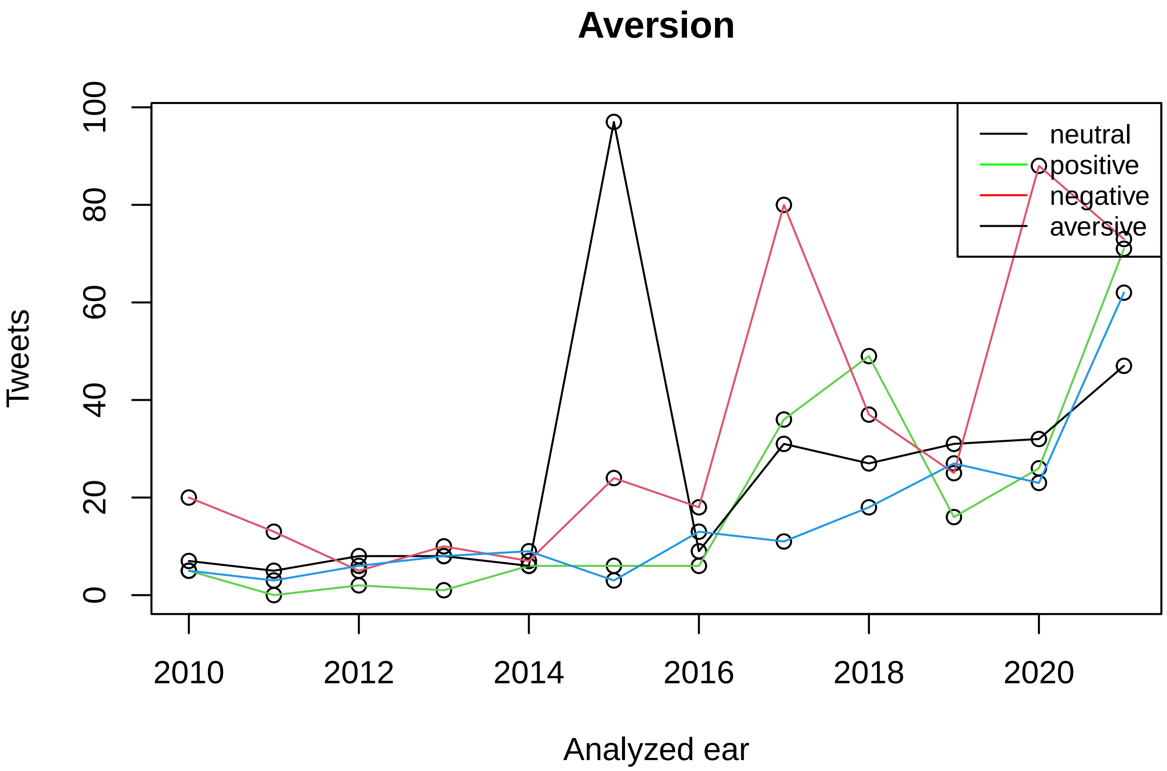


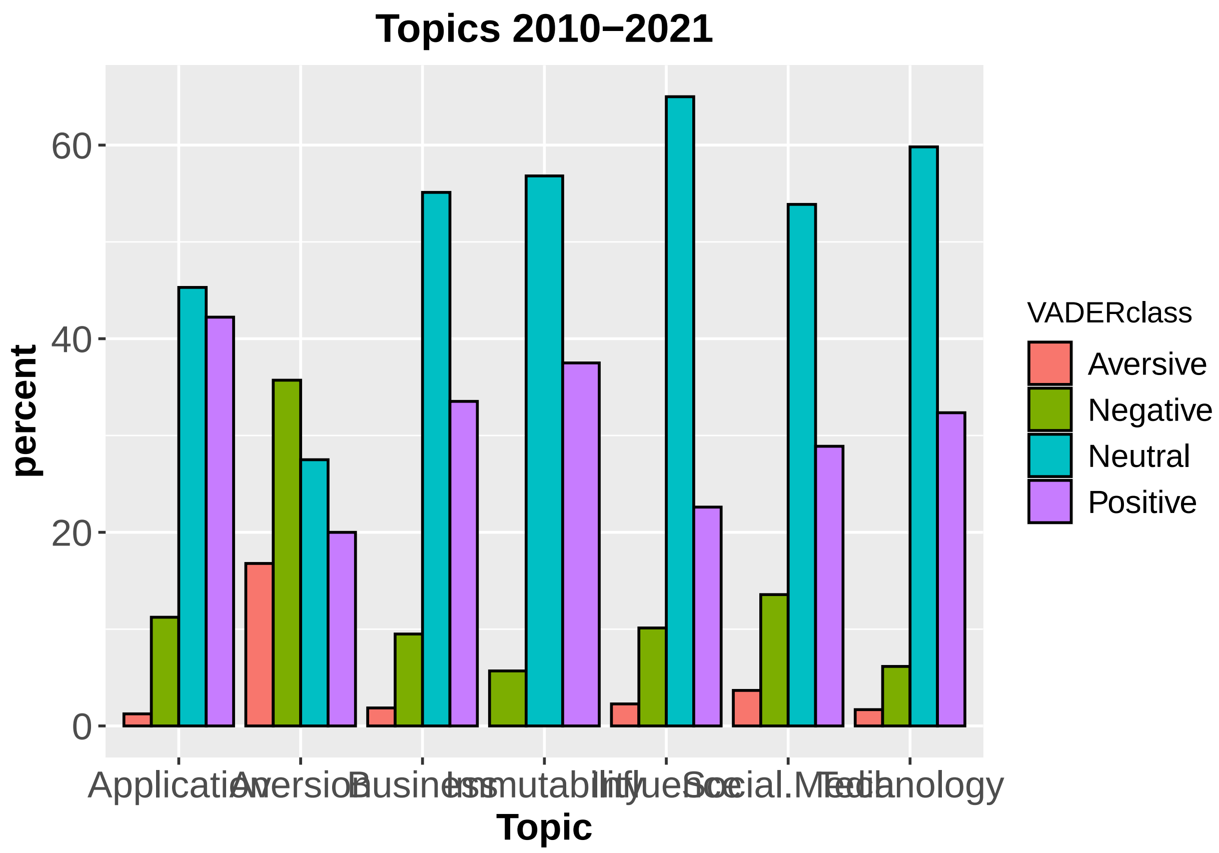

Social.Media
Tweets
0
500
1500
2500
2010
2012
2014
2016
2018
2020
Analyzed ear
neutral
positive
negative
aversive


Technology
neutral
positive
negative
aversive
Tweets
0
200
400
600
2010
2012
2014
2016
2018
2020
Analyzed ear


Influence
Tweets
0
200
400
600
800
2010
2012
2014
2016
2018
2020
Analyzed ear
neutral
positive
negative
aversive


Application
Tweets
0
50
100
150
2010
2012
2014
2016
2018
2020
Analyzed ear
neutral
positive
negative
aversive


Aversion
Tweets
0
20
40
60
80
100
2010
2012
2014
2016
2018
2020
Analyzed ear
neutral
positive
negative
aversive



Motivation: Erwin

Data retrieval &

Data processing: Mathias

Analysis &

Conclusion: Max

Critique: Alina