Annex 5

Data extraction from newspaper's web page and comparison

On the whole set of articles, a total of 145, only for five of them has been possible to write comments straight on the article's web page. The same articles have also been available for comments at Facebook and Twitter. The next table lists those articles commented in the same web page.

Internal id	date	media	n. of comments
29	22/07/18	Independent	76
68	23/07/18	Daily Mail	41
69	22/07/18	Daily Mail	452
107	23/07/18	Daily Express	33
108	23/07/18	Daily Express	20

As shown in the table three newspapers displayed comments in their web pages, each newspaper offered apparently similar commenting systems, but in reality they use different mechanisms, it is interesting to analyse them separately.

Newspaper "Independent" shows user names with bold characters, "Reply" link with red colour, both capture the attention. Two icons "thumb up" and "thumb down" with the number of likes in the middle. The number of likes is the algebraic sum of positive and negative score.

Newspaper "Daily Mail" shows the number of likes and the number of dislikes separately.

Newspaper "Daily Express" calculates the number of likes as algebraic sum of positive and negative score, moreover allows to share or report the comment. Some users make available their avatar picture.

An important finding common to the three newspapers web site commenting system is related to anonymity, users account do not really disclose personal information, users identity cannot be validated by any mean, even for those few users with real faces as avatars. Moreover the censorship activity removed some avatars and comments.

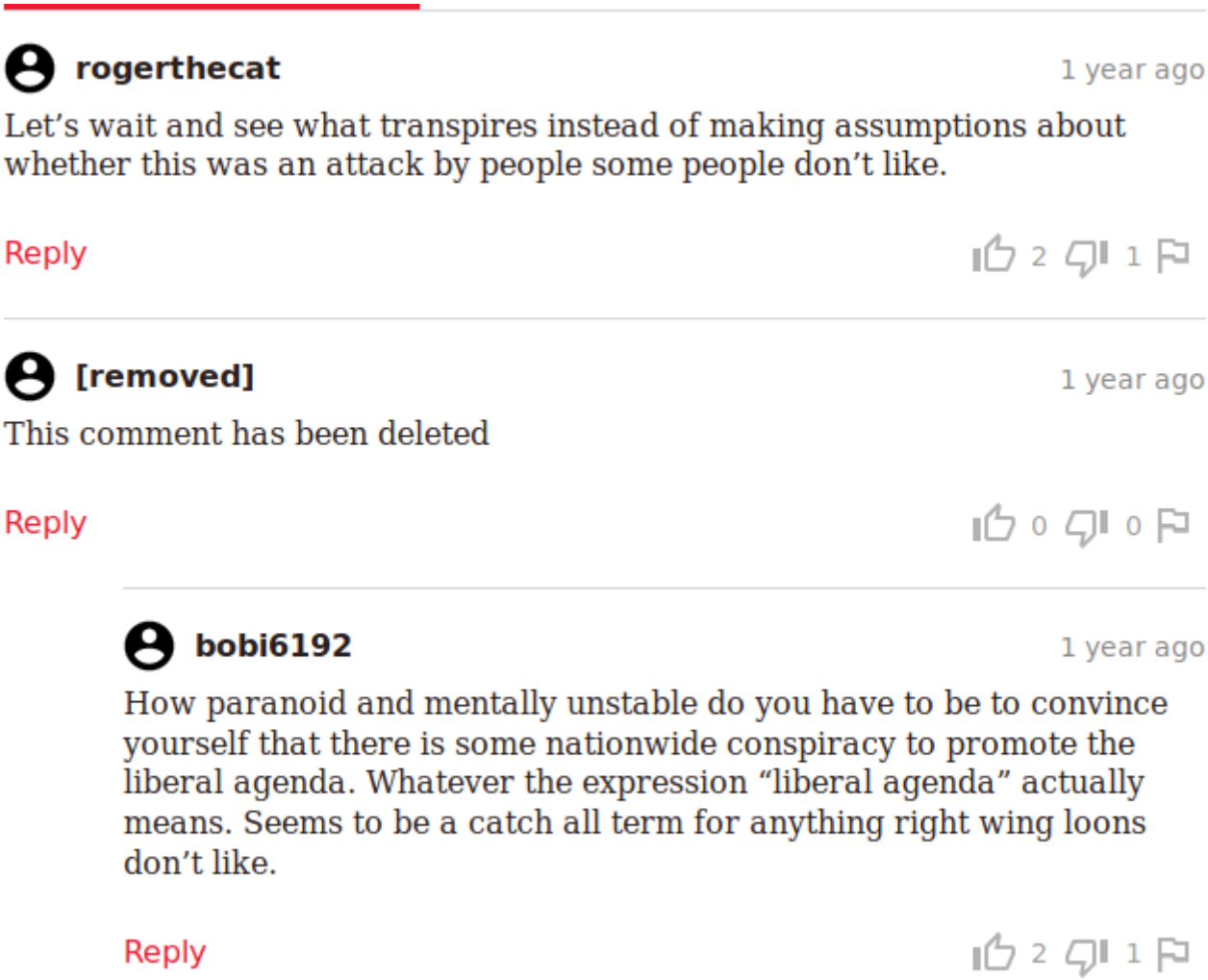
The three newspapers do not make available Application Program Interfaces (APIs), a third part service, newsapi.org provides APIs that get news from Independent and Daily Mail. There are no APIs able to read comments.

In order to extract the network diagram and the set of comments from newspapers web sites it has been necessary to develop specific software programs that implement web scraping techniques.

The first step analyses the Hypertext Markup Language (HTML) code behind the web pages for searching patterns that allow to capture data. The syntax used is XPath, it allows to find elements in the HTML code.

Independent newspaper web comments (internal id 29)

The next picture shows the comment section, user names are not obfuscated because they do not disclose real identities.



The elements to capture are the user name, the comment text and the position to determine whether is a direct comment or a reply to an existing comment.

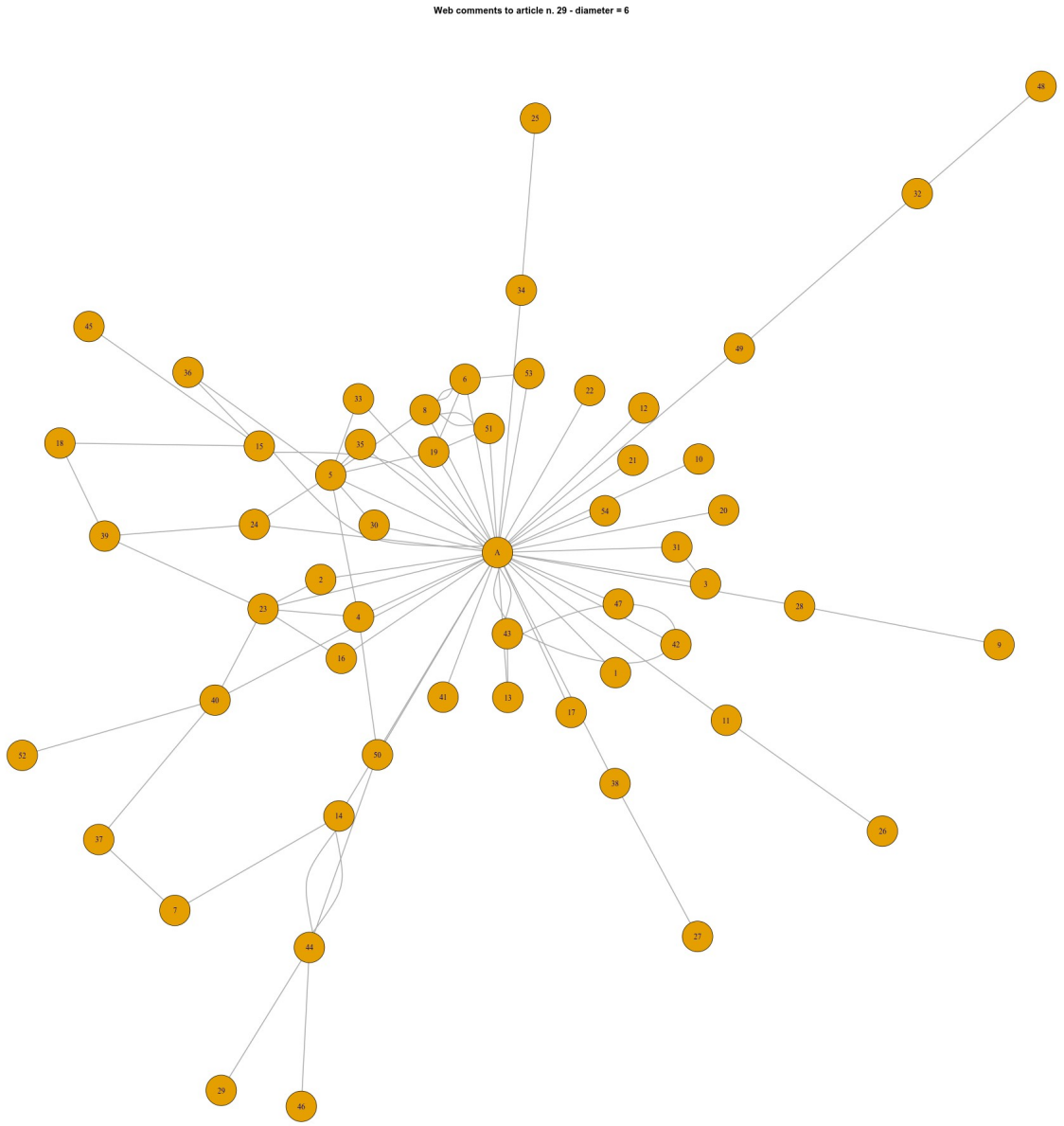
The analysis ot HTML code resulted in the following patterns:

- user name is coded as span element with attribute class = "user"
- comments are coded as div element with attribute class = "comment-text"
- elements "data-comment-id" and "class" indicate the level of reply

The next illustration shows the code in R programming language for extracting data.

```
library(rvest)
library(dplyr)
library(grr)
library(igraph)
html_doc <- read_html(paste0(dir_tests,art_num,'.html'), encoding = 'UTF-8')
node_comments <- html_nodes(html_doc, xpath = "//div[@class='comment card']")
len_df = length(node_comments)
df <- data.frame(
       'un' = character(len_df), # user name
       'dd' = character(len_df), # data-comment-id
       'cn' = character(len_df), # class name
       'nl' = integer(len_df),   # number of likes
       'nd' = integer(len_df),   # number of dislikes
       'cm' = character(len_df), # comments text
       stringsAsFactors=FALSE
)
for (ind in 1:len_df) {
       df[ind,2] <- html_attr(html_nodes(node_comments[ind], xpath = "."),'data-comment-
id')
       df[ind,3] <- html_attr(html_nodes(node_comments[ind], xpath = ".."),'class')
}
df$un <- html_text(html_nodes(node_comments, xpath = "//span[@class='user']"))
df$nl <- as.integer(html_text(html_nodes(node_comments, xpath = "//span[@class='vote-
count'][1]")))
df$nd <- as.integer(html_text(html_nodes(node_comments, xpath = "//span[@class='vote-
count'][2]")))
df$cm <- html_text(html_nodes(node_comments, xpath =
"//div[@class='comment-text']/text()"))
re <- 1
for (ind in 1:len_df) {
       if (df[ind,1] == '[removed]') {
       df[ind,1] <- paste0('removed_',re)
       re <- re + 1
       }
}
df[is.na(df)] <- ''
df$rp <- ''
for (ind in 1:len_df) { if ( ! df[ind,2] == '' ) { df[ind,7] <- newspaper } }
for (ind in 1:len_df) {
       if ( df[ind,7] == '' & df[ind,3] == 'replies-1' ) {
       un <- df[ind - 1,1]
       wnd <- ind
       while ( df[wnd,7] == '' ) {
              if ( df[wnd,3] == 'replies-1' ) { df[wnd,7] <- un }
              wnd <- wnd + 1
       }
       ind <- wnd
       }
}
for (ind in 1:len_df) {
       if ( df[ind,7] == '' & df[ind,3] == 'replies-2' ) {
       un <- df[ind - 1,1]
       wnd <- ind
       while ( df[wnd,7] == '' ) {
              if ( df[wnd,3] == 'replies-2' ) { df[wnd,7] <- un }
              wnd <- wnd + 1
       }
       ind <- wnd
       }
}
fqun <- as.data.frame(table(df$un)) # get unique records of user names and calculates
frequencies
fqun <- mutate(fqun, id = rownames(fqun)) # adds column id and populates it
colnames(fqun)[1] <- 'name'
df$un <- as.factor(df$un)
dict <- grr::matches(fqun$name, df$un)
dict_sorted <- dict[order(dict[,2]),]
df <- cbind(df,dict_sorted[,1])
colnames(df)[8] <- 'id_un'
df$id_rp <- match(df$rp, fqun$name, 0)
df$id_rp[df$id_rp==0] <- 'A'
ph <- vector('character');
len <- nrow(df)
for (row in 1:len){
       ph <- c(ph,c(df[row,9],df[row,8]))
}
gr <- graph(ph,directed = FALSE)
```

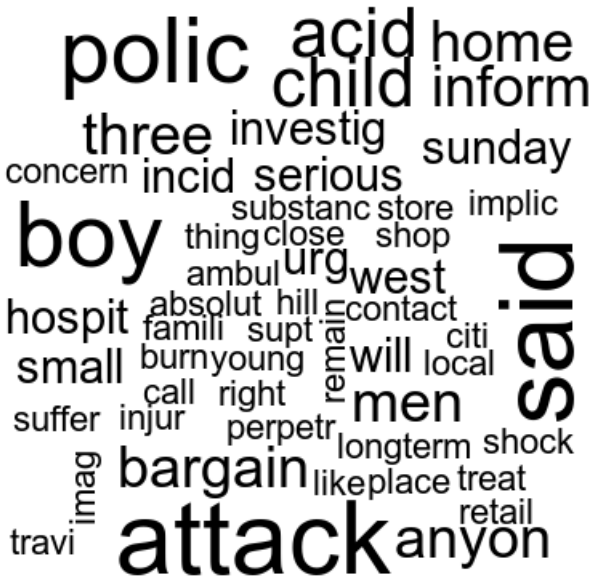
The next illustration shows the network diagram



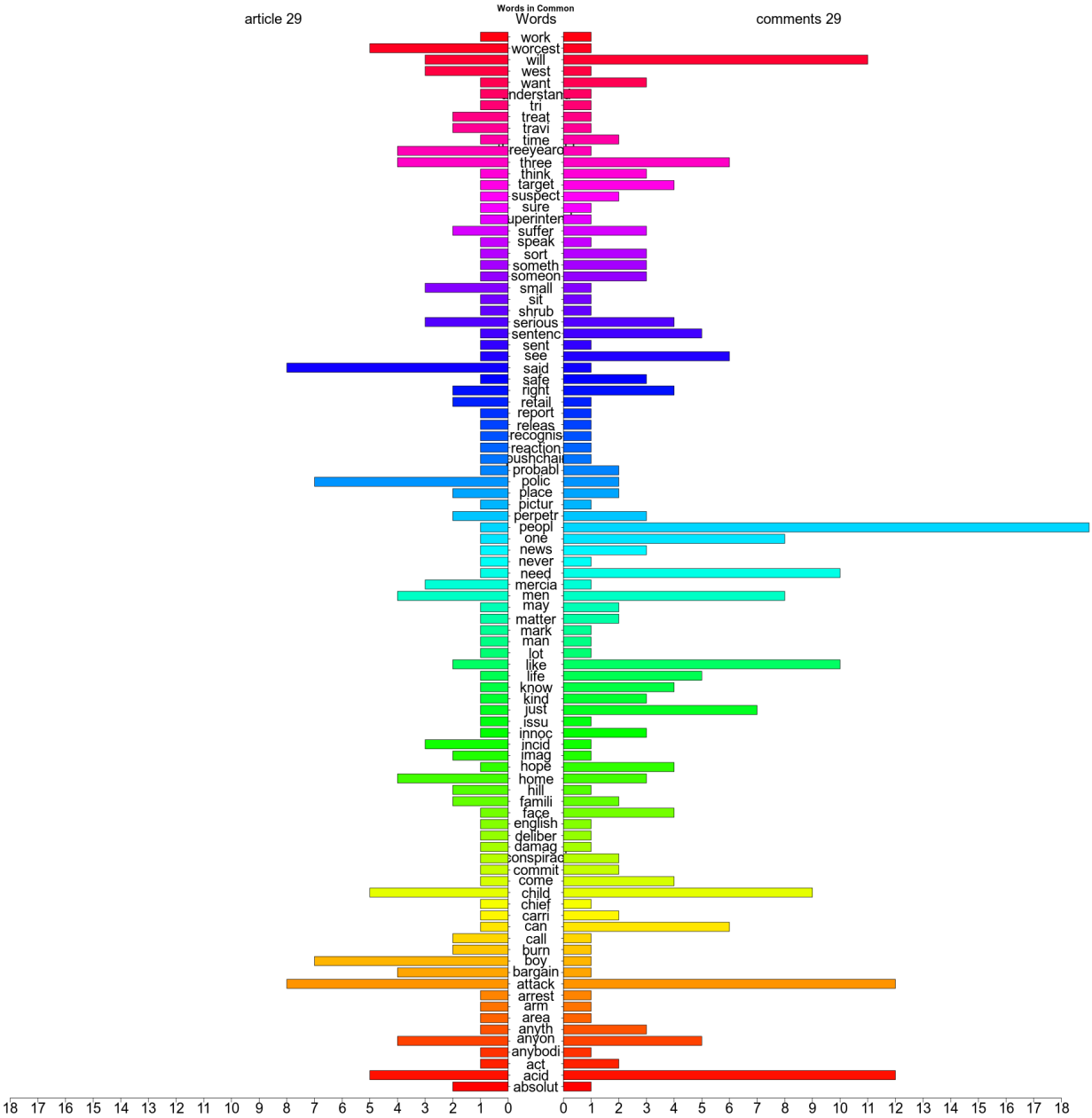
The central node is the article's author. All names are replaced by numbers. The network diameter is 6.

Now proceed with the comparison between the article's text and the set of comments.

The next illustrations show the word cloud representations of the article and the comments with a minimal frequency of two words and the pyramid diagram of common words.


polic
acid
home
child
inform
three
investig
sunday
concern
incid
serious
substanc
store
implic
boy
thing
close
shop
ambul
urg
west
said
hospit
absolut
hill
contact
famili
supt
remain
citi
small
burn
young
will
local
call
right
men
suffer
injur
perpetr
longterm
shock
imag
bargain
like
place
treat
retail
travi
attack
anyon


comme
person
street
allow
delet
minor
commit
point
remov
vat
hate
sick
serv
way
pay
sentenc
flag
death
given
done
jail
live
eye
drop
crime
good
get
anoth
know
murder
catch
honest
time
act
idiot
carri
upon
think
acid
joke
old
aliv
kind
sort
target
look
stori
throw
human
must
promot
come
polic
post
utter
let
hope
pain
life
arent
victim
men
sinc
sale
even
warn
walk
innoc
larg
ive
crimin
perpetr
agenda
feel
came
instead
total
automat
toler
drive
news
hour
paranoid
want
home
dealt
web
care
convict
read
tab
cours
defend
assumpt
countri
agre
three
long
liber
probabl
one
keep
usual
face
ason



The R programming code is the same shown in annex 4 titled “Comparison between article’s text and comments”

Jaccard similarity value calculated is 0.33

Daily Mail newspaper web comments (internal id 68)

The next picture shows the comment section, user names are not obfuscated because they do not disclose real identities.

The elements to capture are the user name, the comment text and the position to determine whether is a direct comment or a reply to an existing comment.

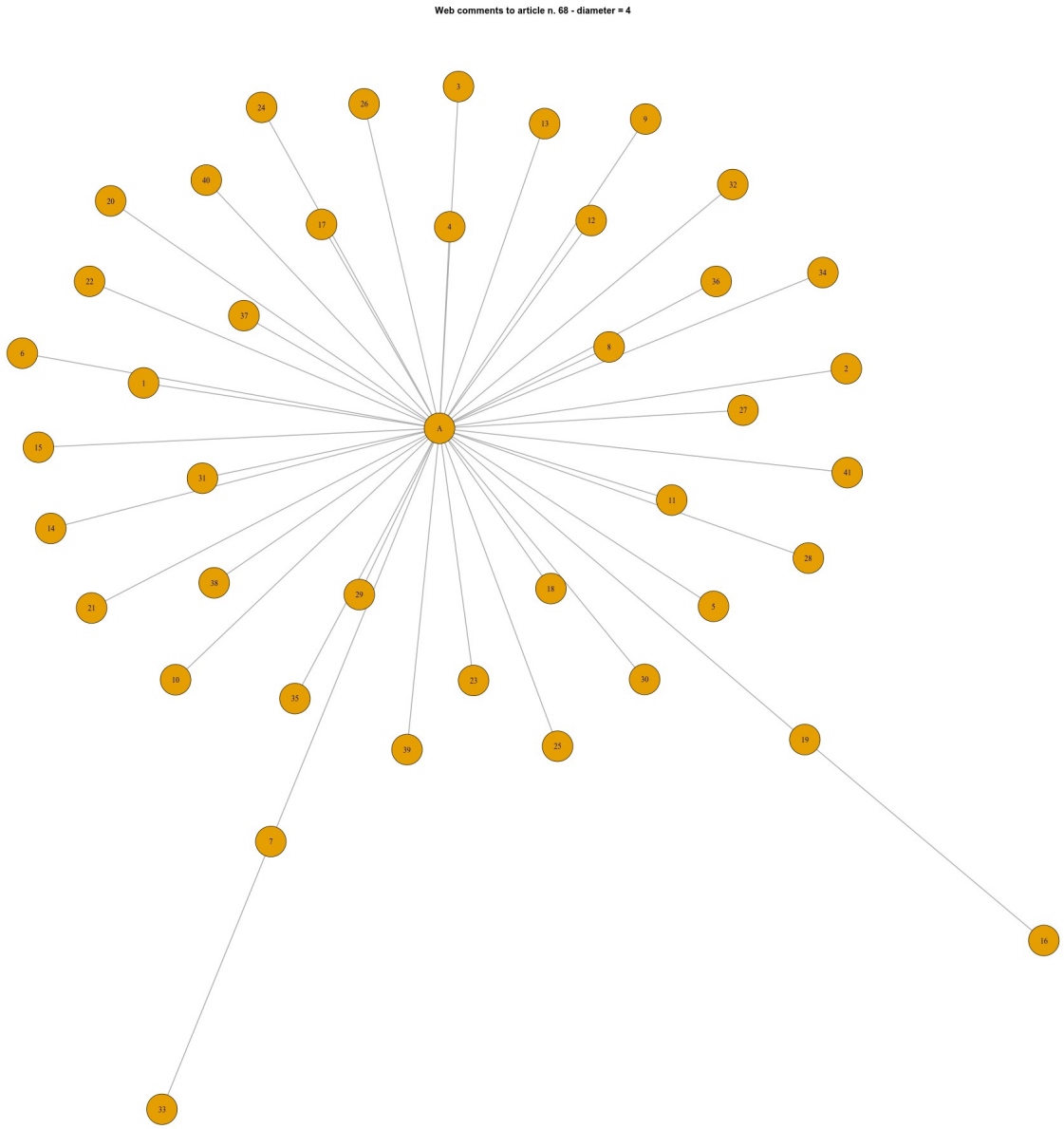
The analysis ot HTML code resulted in the following patterns:

- user name is coded as p element with attribute class = “user-info”
- comments are coded as p element with attribute class starting with “comment” or “reply”
- elements after elements p with attribute class = “user-info” indicate the level of reply

The next illustration shows the code in R programming language for extracting data.

```
library(rvest)
library(dplyr)
library(grr)
library(igraph)

html_doc <- read_html(paste0(dir_tests,art_num,'_comments.html'), encoding = 'UTF-8')
node_comments <- html_nodes(html_doc, xpath = "//div[starts-with(@id,'comment-')]")
len_df = length(node_comments)

# user names
un <- html_text(html_nodes(node_comments,xpath="//p[@class='user-info']/a/text()"))
# comments text
cm <- html_text(html_nodes(node_comments,xpath="//p[starts-with(@class,'comment') or
starts-with(@class,'reply')]/text()"))
# number of likes
nl <- as.integer(html_text(html_nodes(node_comments,xpath="//div[@class='rate-up']/
following-sibling::div[@class='rating-button-inline']/text()")))
# number of dislikes
nd <- as.integer(html_text(html_nodes(node_comments,xpath="//div[@class='rate-down']/
following-sibling::div[@class='rating-button-inline']/text()")))
# type of comment
tc <- html_text(html_nodes(node_comments,xpath="//p[@class='user-info']/following-
sibling::p/@class"))
tc <- gsub('.*comment.*','comment',tc)
tc <- gsub('.*reply.*','reply',tc)

df <- data.frame(
       'un' = character(len_df),
       'nl' = integer(len_df),
       'nd' = integer(len_df),
       'cm' = character(len_df),
       'tc' = character(len_df),
       stringsAsFactors=FALSE
)

df$un <- un
df$nl <- nl
df$nd <- nd
df$cm <- cm
df$tc <- tc

fqun <- as.data.frame(table(df$un))     # get unique records of user names and
calculates frequencies
fqun <- mutate(fqun, id = rownames(fqun))      # adds column id and populates it
colnames(fqun)[1] <- 'name'
df$un <- as.factor(df$un)  # otherwise grr::matches fires error
dict <- grr::matches(fqun$name, df$un)
dict_sorted <- dict[order(dict[,2]),]
df <- cbind(df,dict_sorted[,1])
colnames(df)[6] <- 'id_un'


# block extract comments ---------------------------------------------------
fcon <- file(paste0(dir_tests,art_num,'_comments.txt'))
writeLines(toString(df$cm), fcon)
close(fcon)

# block net graph ----------------------------------------------------
ph <- vector('character');
len <- nrow(df)
for (row in 1:len){
       if (df[row,5] == 'comment') { ph <- c(ph,c('A',df[row,6])) }
       if (df[row,5] == 'reply') {
       uid <- df[row-1,6]
       while (df[row,5] == 'reply') {
              ph <- c(ph,c(uid,df[row,6]))
              row <- row + 1
       }
       }
}
gr <- graph(ph,directed = FALSE)
```

The next illustration shows the network diagram

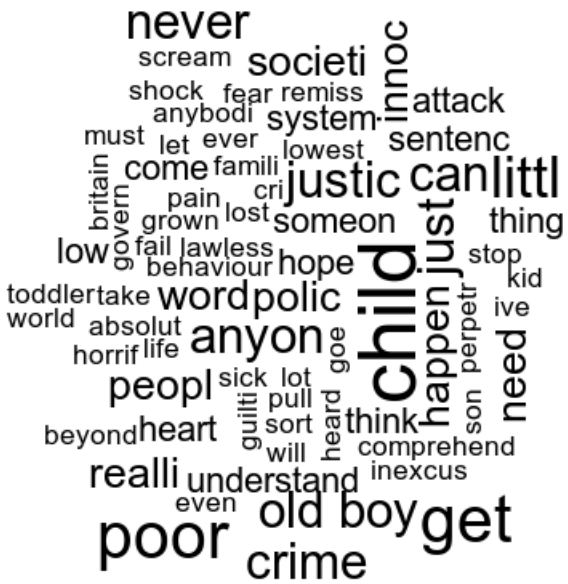


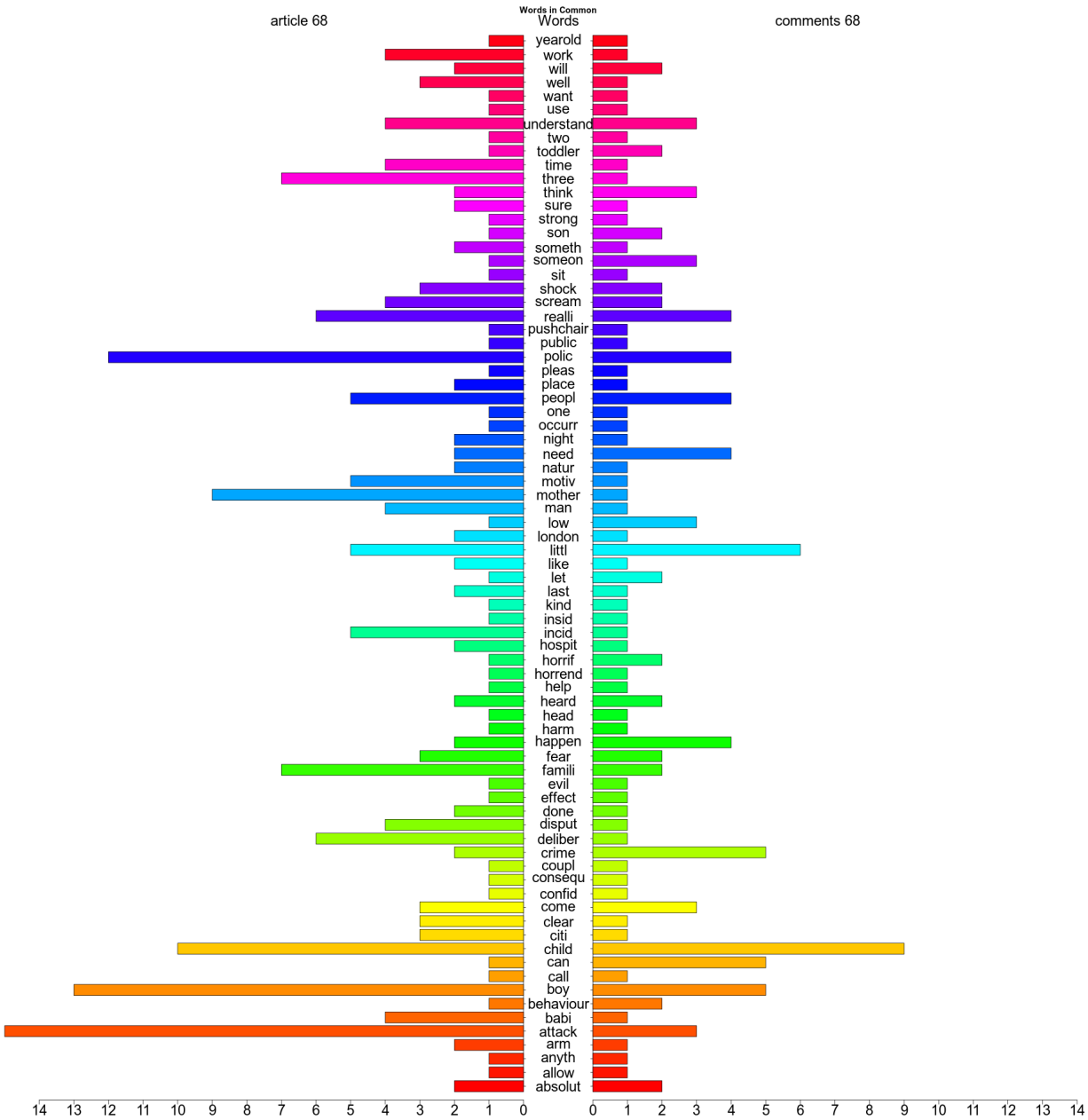
The central node is the article's author. All names are replaced by numbers. The network diameter is 4.

Now proceed with the comparison between the article's text and the set of comments.

The next illustrations show the word cloud representations of the article and the comments with a minimal frequency of two words and the pyramid diagram of common words.


west understand investig describ seen
quit toward
captur scream boy home
commit detect dknow
well sure around
someth citi injuri area
move told shop bargain like park
absolut shock
heard juli away threeyearold believ
ambul
will acid men supt moment pink mother afternoon
arm suggest place
part time travi mercia fear suspicion need target
burn hospit bst motiv imag issu ran
link
show appeal mark put crime 2018 come
short think conspiraci wit
safeti treat look done smell outsid taken
face deliber store littl communiti
babi london incid still three way grievous
saturday
make attack local bodili clear hous
chief
safe man substanc last famili
longterm
natur term night remain howev


never
scream societi innoc
shock fear remiss
anybodi system attack
must let ever lowest sentenc
britain come famili justic canlittl
govern pain cri
grown lost someon just thing
low fail lawless
behaviour hope child stop
kid
toddlertake word polic happen ive
world absolut anyon perpetr need
horrif life goe
peopl sick lot
guilti pull heard son
beyond heart sort think
will comprehend
realli understand inexcus
even
poor old boy get
crime



The R programming code is the same shown in annex 4 titled “Comparison between article’s text and comments”

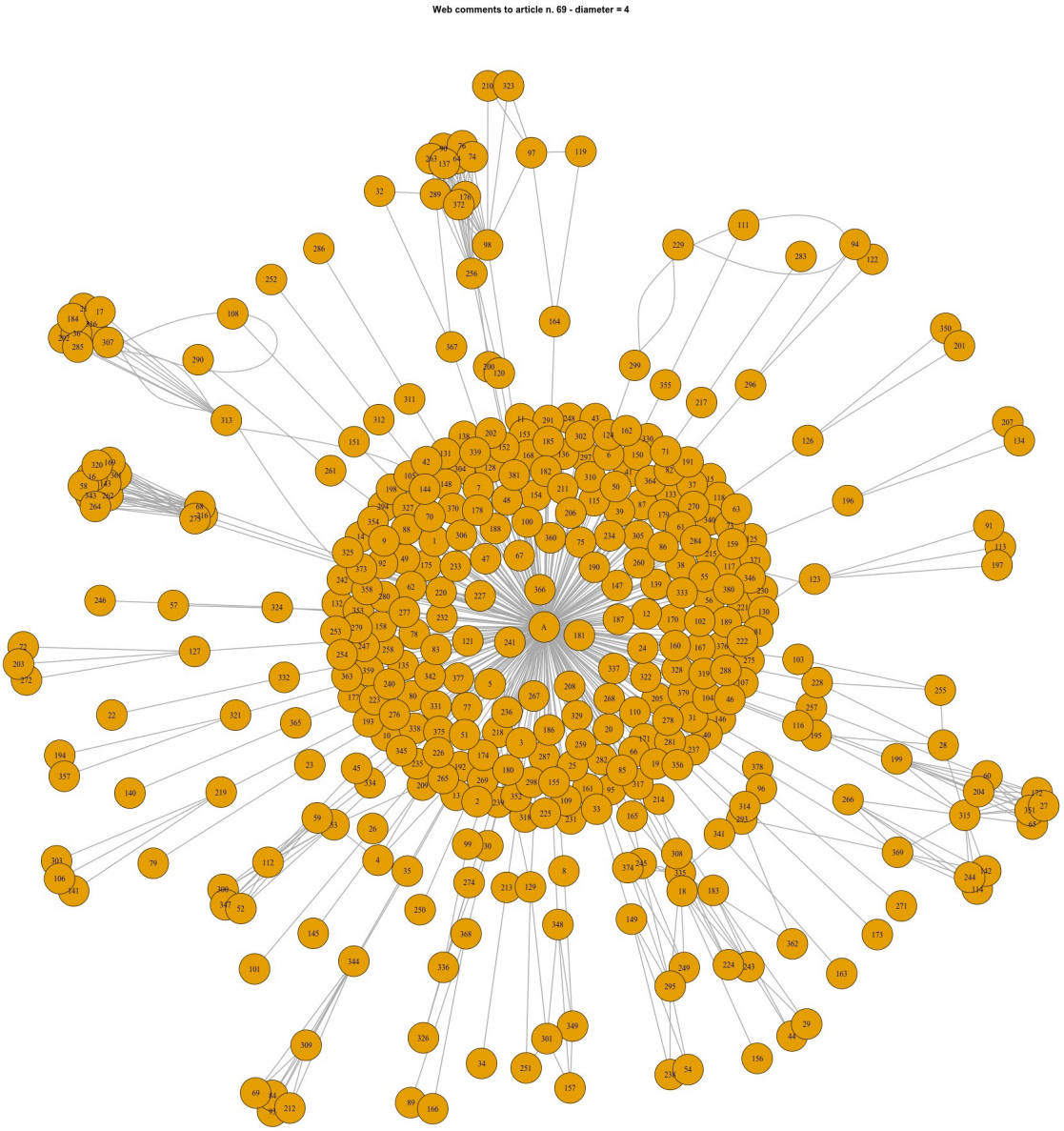
Jaccard similarity value calculated is 0.32

Daily Mail newspaper web comments (internal id 69)

The next picture shows the comment section, user names are not obfuscated because they do not disclose real identities.

The analysis ot HTML code and R programming code are the same as previous web comments (internal id 68)

The next illustration shows the network diagram



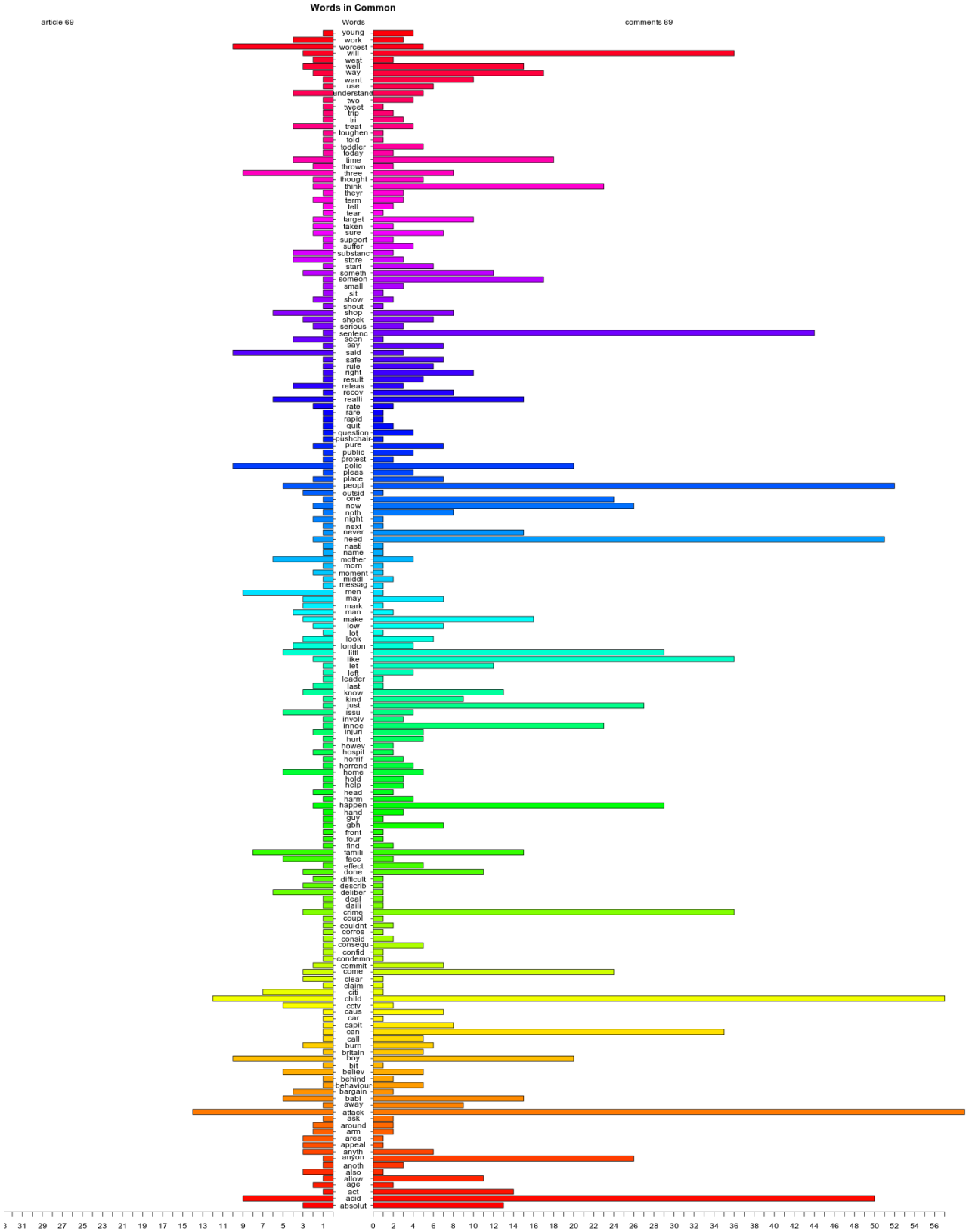
The central node is the article's author. All names are replaced by numbers. The network diameter is 4.

Now proceed with the comparison between the article's text and the set of comments.

The next illustrations show the word cloud representations of the article and the comments with a minimal frequency of two words and the pyramid diagram of common words.


conspiraci
member
short
scream
known
believ
worcest
come
around
night
area
outsid
work
remain
taken
longterm
bst
know
said
need
heard
took
age
anyth
substanc
think
now
cctv
shock
pure
wit
thug
show
investig
grievous
incid
well
juli
safest
toward
natur
commit
local
chief
deliber
likesure
hurl
appeal
make
wolverhampton
ran
difficult
low
hospit
realli
moment
thought
releas
move
disput
captur
way
child
issu
will
travi
arm
littl
term
39yearold
also
retail
link
last
serious
thrown
place
imag
babi
suspicion
head
rate
bodili
park
hill
time
arrest
men
footag
west
absolut
threeyearold
look
man
happen
yesterday
burn
superintend


unthink
hospitbelief
offenc
shame
judg
vehicl
consid
realiti
barbar
andrea
dna
sick
love
mind
free
avail
countri
will
mess
lost
speedi
brought
reloc
deterr
give
lawless
wish
charg
enough
can
sometim
light
joke
may
home
tax
wouldnt
young
peapod
worri
place
fail
guilti
core
root
doesnt
arent
risk
save
youv
drug
run
agre
need
couldnt
babi
insid
caught
comment
fit
lock
word
soft
that
shouldnt
time
want
full
put
show
intent
establish
prevent
yet
sea
ask
pay
germani
leadership
heart
system
court
call
visitor
media
assault
allow
duelesser
injuri
room
uninhabit
though
kill
legal
receiv
today
public
weak
west
britain
adult
forc
ive
choic
probabl
gbh
imprison
opinion
continu
sever
caus
catch
reason
lad
anoth
seem
what
trull
real
tri
tougher
parti
feel
fear
job
quit
play
limit
alright
rope
cover
left
still
kept
whatev
grip
involv
carri
add
asap
protest
treat
bloodi
idiot
sane
cribest
mrs
massiv
right
send
agreement
ever
support
crisi
culprit
toni
small
found
harm
sort
feet
rid
polit
two
listen
wrong
believ
nobodi
hous
behaviour
girl
far
recoveri
labour
quick
lowest
sake
devast
beat
kind
obvious
eye
issu
easi
bless
shock
expens
mother
god
mani
justic
boy
cctv
step
film
person
sign
physic
answer
crazi
horrend
lack
wont
trip
human
reinstat
lenient
work
without
middl
face
liber
increas
repeat
way
blame
think
low
thrown
nation
old
keep
usual
screen
trump
around
unabl
expect
stay
target
vile
elis
current
100
much
now
minimum
respons
suffer
control
demand
either
possibl
throw
deter
meal
head
amount
violenc
rule
matter
shoot
island
especi
set
bad
protect
world
jail
crime
bleach
inhuman
stuff
america



The R programming code is the same shown in annex 4 titled “Comparison between article’s text and comments”

Jaccard similarity value calculated is 0.36

Daily Express newspaper web comments (internal id 107)

The next picture shows the comment section, user names are not obfuscated because they do not disclose real identities.

The elements to capture are the user name, the comment text and the position to determine whether is a direct comment or a reply to an existing comment.

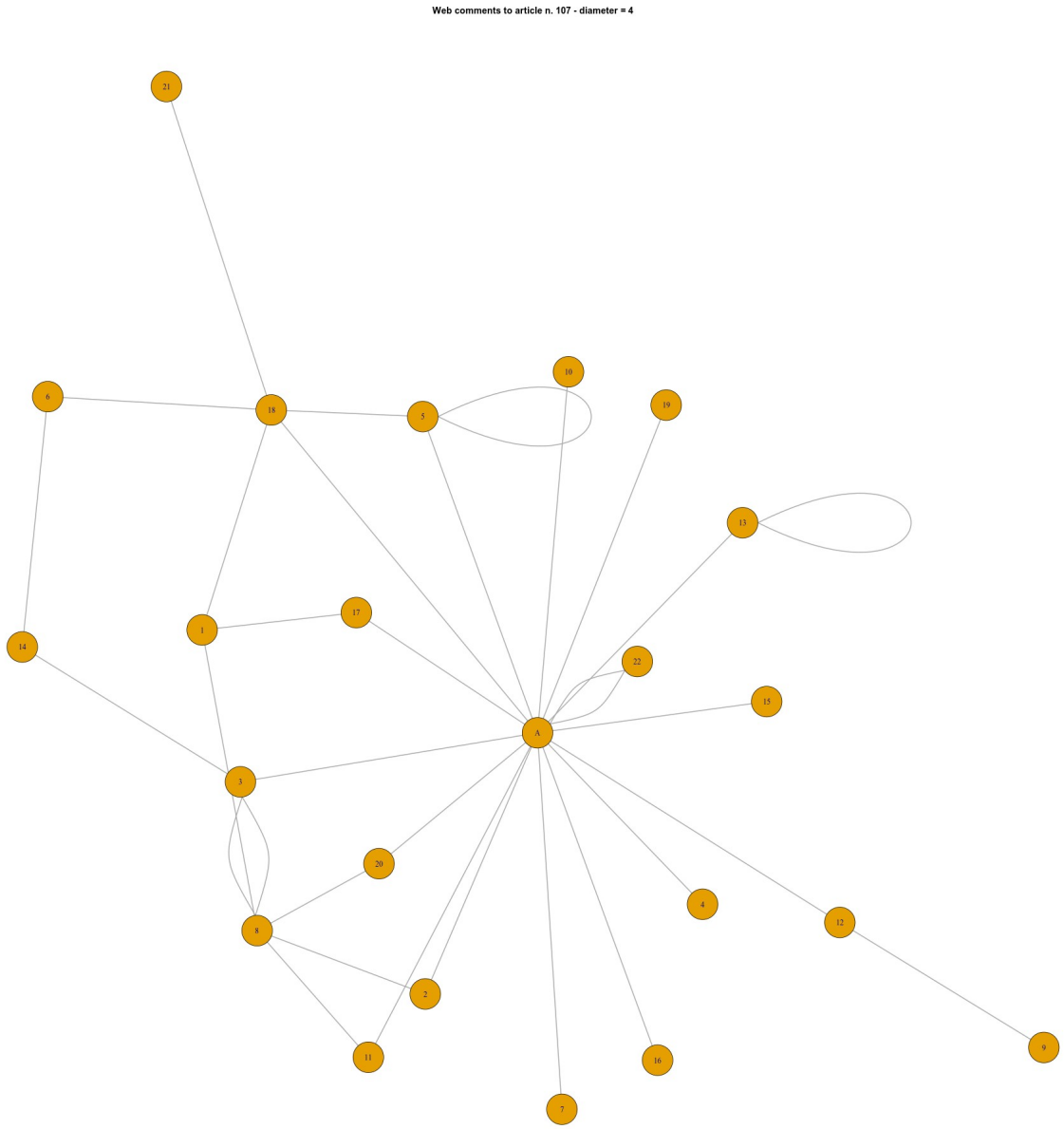
The analysis ot HTML code resulted in the following patterns:

- user name is coded as span element with attribute data-spot-im-class = “message-username”
- comments are coded as div element with attribute data-spot-im-class = “message-text”
- the level of reply is coded in a extremely complicated way, since the number of comments is low therefore has been more convenient to fill this information manually based on a built table

The next illustration shows the code in R programming language for extracting data.

```
library(rvest)

html_doc <- read_html(paste0(dir_tests,art_num,'_comments.html'), encoding = 'UTF-8')
un <- html_text(html_nodes(html_doc, xpath = "//span[@data-spot-im-class='message-
username']/text()"))
cm <- html_text(html_nodes(html_doc, xpath = "//div[@data-spot-im-class='message-text']/
text()"))

len_df = length(un)

df <- data.frame(
       # user name
       'un' = character(len_df),
       # comments text
       'cm' = character(len_df),
       stringsAsFactors=FALSE
)

df$un <- un
df$cm <- cm

write.csv(df,paste0(dir_tests,art_num,'_comments.csv'))

# then manual fill

# block extract comments ---------------------------------------------------
fcon <- file(paste0(dir_tests,art_num,'_comments.txt'))
writeLines(toString(df$cm), fcon)
close(fcon)

# block net graph ----------------------------------------------------

library(dplyr)
library(grr)

dir_tests <- '/home/oreste/Downloads/'
art_num <- 107
newspaper <- 'Express'

df <- read.csv(paste0(dir_tests,art_num,'_graph.csv')) # filled manually

# un = user name
# cm = comments text
# nl = number of likes
# rp = replies to (character)
# le = level of reply (1 to 4)

fqun <- as.data.frame(table(df$un))     # get unique records of user names and
calculates frequencies
fqun <- mutate(fqun, id = rownames(fqun))      # adds column id and populates it
colnames(fqun)[1] <- 'name'
df$un <- as.factor(df$un)  # otherwise grr::matches fires error
dict <- grr::matches(fqun$name, df$un)
dict_sorted <- dict[order(dict[,2]),]
df <- cbind(df,dict_sorted[,1])
colnames(df)[5] <- 'id_un'

df$id_rp <- match(df$rp,fqun$name, 0)

write.csv(df,paste0(dir_tests,art_num,'_igraph.csv'))

#-------------------------------------------------------------------
library(igraph)

dir_tests <- '/home/oreste/Downloads/'
art_num <- 107
newspaper <- 'Express'

df <- read.csv(paste0(dir_tests,art_num,'_igraph.csv'))
df[df==0]<-'A'

ph <- vector('character');
len <- nrow(df)
for (row in 1:len){ ph <- c(ph,c(df[row,7],df[row,6])) }

gr <- graph(ph,directed = FALSE)
```

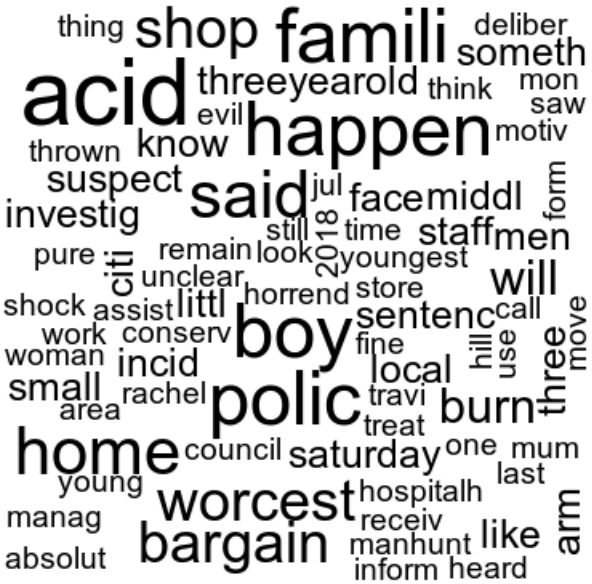
The next illustration shows the network diagram



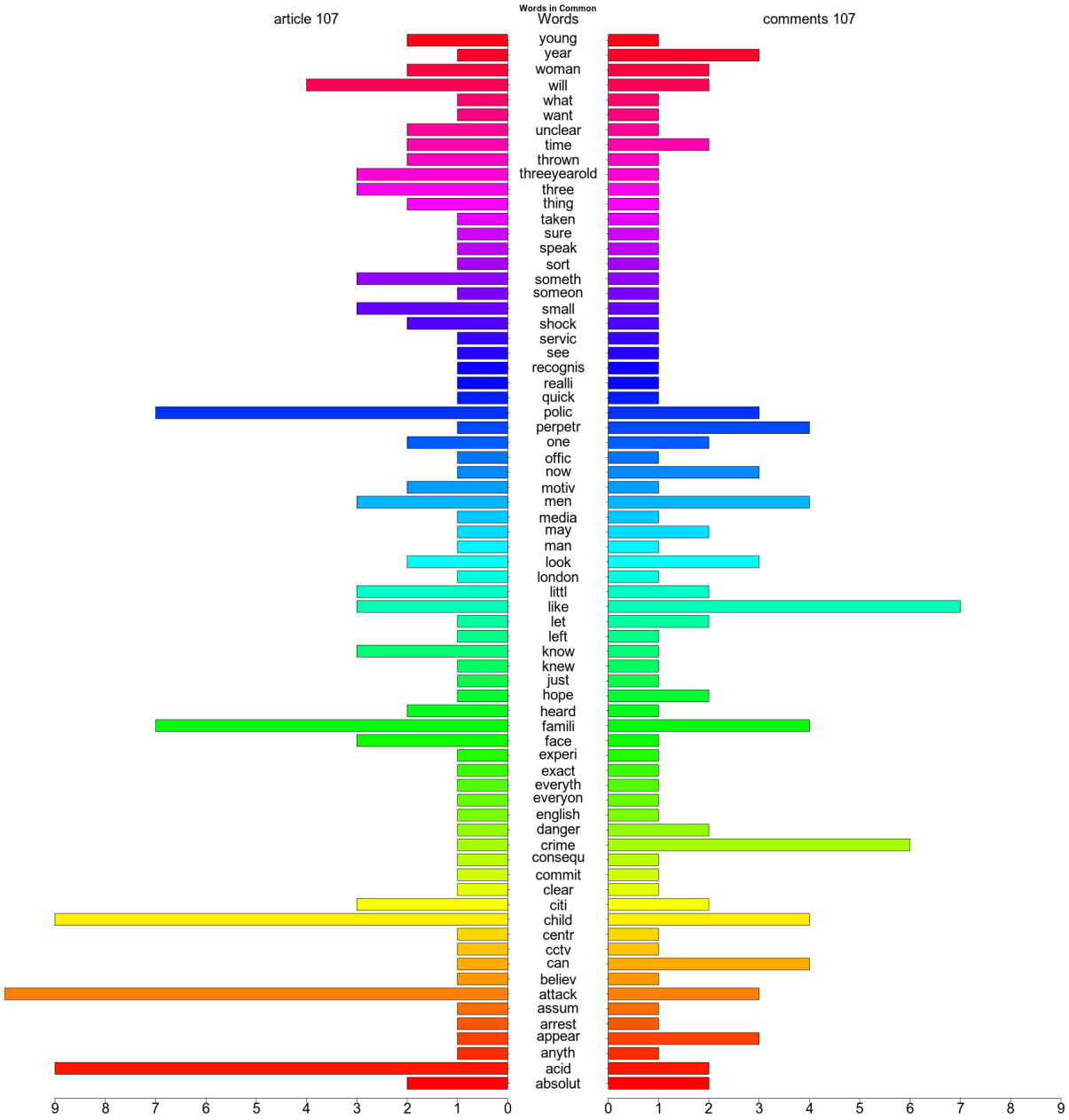
The central node is the article's author. All names are replaced by numbers. The network diameter is 4.

Now proceed with the comparison between the article's text and the set of comments.

The next illustrations show the word cloud representations of the article and the comments with a minimal frequency of two words and the pyramid diagram of common words.


thing
shop
famili
deliber
someth
acid
threeyearold
think
mon
saw
evil
happen
motiv
thrown
know
suspect
said
jul
facemiddl
form
investig
still
time
staffmen
2018
pure
citi
remain
look
youngest
unclear
will
shock
assist
littl
horrend
store
sentenc
call
work
conserv
boy
fine
move
woman
incid
hill
use
three
small
rachel
local
area
polic
travi
burn
treat
home
council
saturday
one
mum
young
last
worcest
hospitalh
manag
receiv
arm
bargain
like
absolut
manhunt
inform
heard


person
anoth
tan
allow
dont
arthurincorrect
year
england
read
like
mart
white
huge
free
littl
life
now
can
charg
citi
prison
letacid
destroy
inch
europ
attack
yet
eastern
glad
1st
report
differ
syrian
prosecut
countri
old
level
good
appear
open
will
run
may
murder
danger
tri
time
woman
mention
press
hope
ever
court
famili

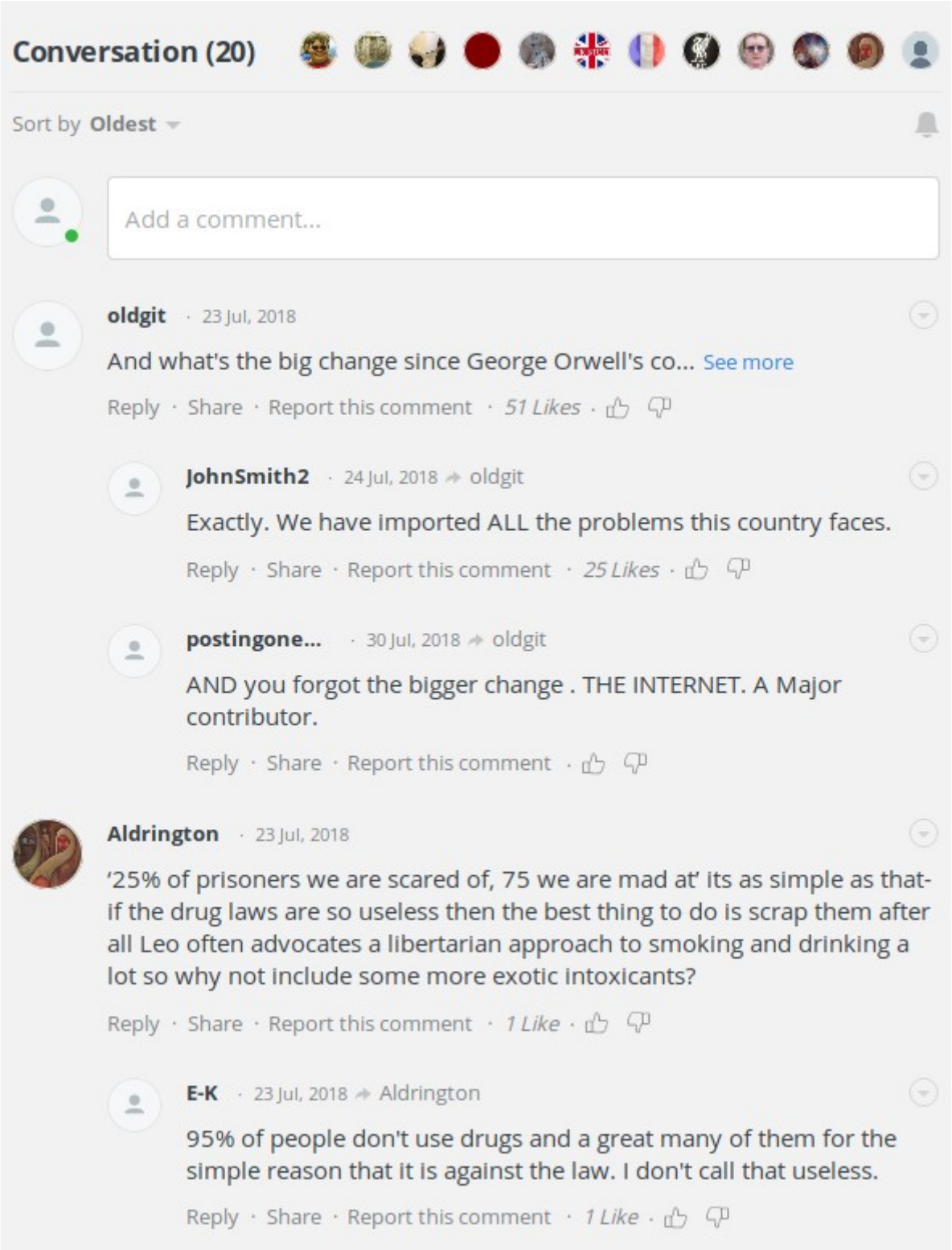


The R programming code is the same shown in annex 4 titled “Comparison between article’s text and comments”

Jaccard similarity value calculated is 0.28

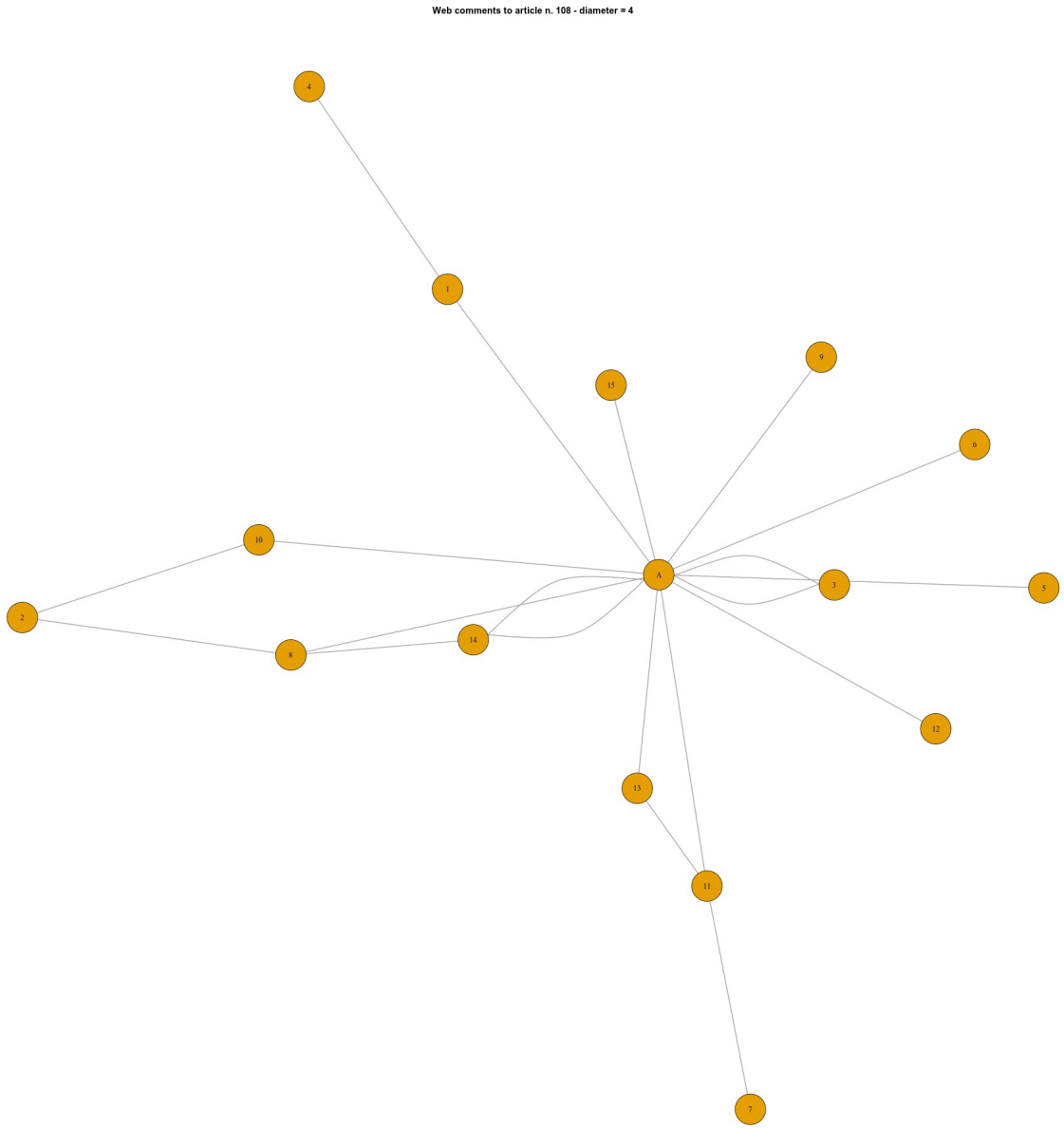
Daily Express newspaper web comments (internal id 108)

The next picture shows the comment section, user names are not obfuscated because they do not disclose real identities.



The analysis ot HTML code and R programming code are the same as previous web comments (internal id 107)

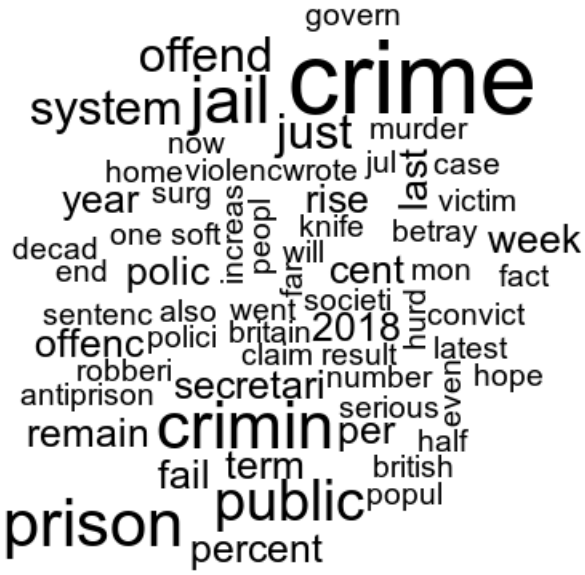
The next illustration shows the network diagram



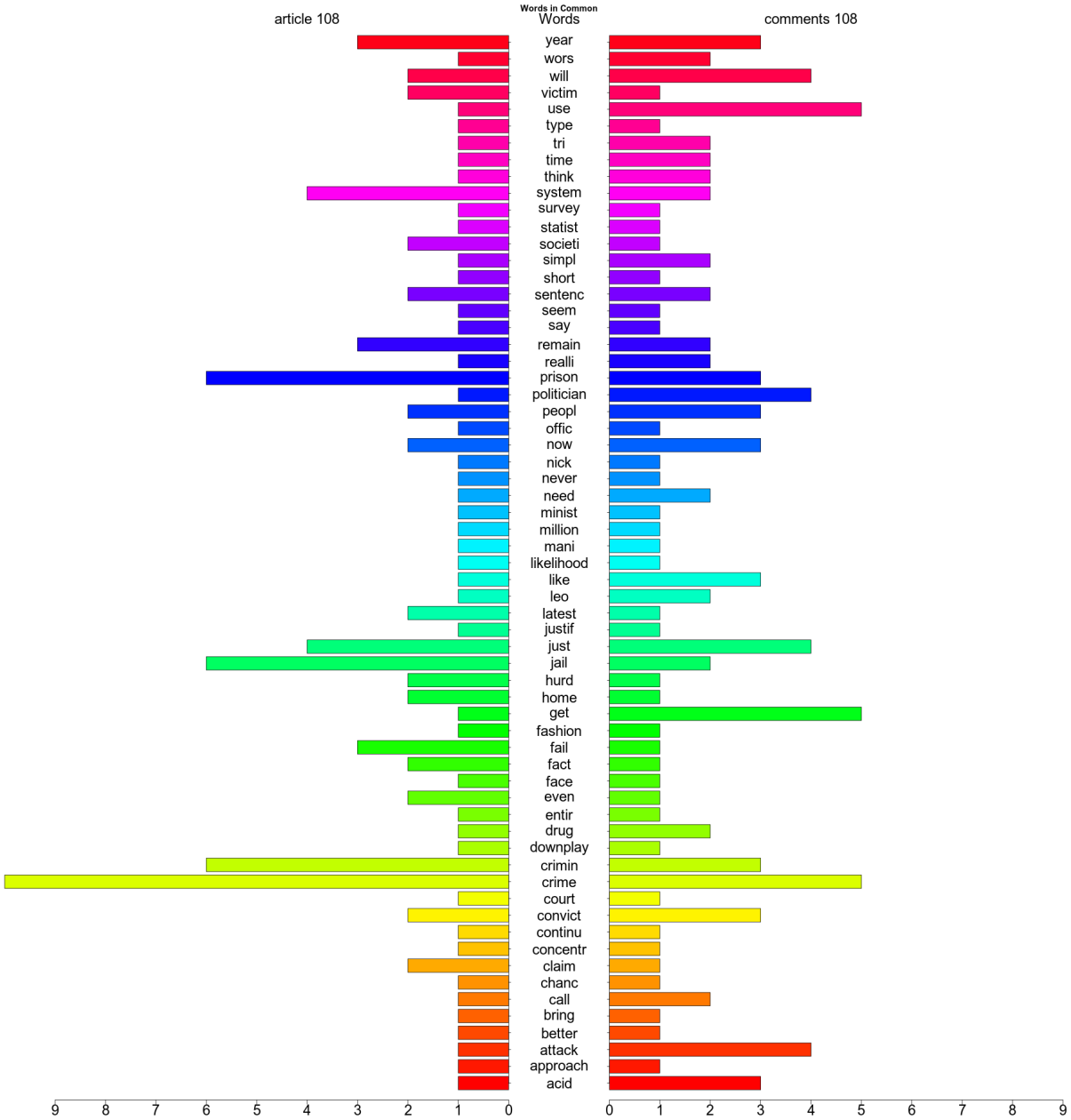
The central node is the article's author. All names are replaced by numbers. The network diameter is 4.

Now proceed with the comparison between the article's text and the set of comments.

The next illustrations show the word cloud representations of the article and the comments with a minimal frequency of two words and the pyramid diagram of common words.


govern
offend
crime
system
jail
just
murder
now
home
violenc
wrote
jul
case
year
surg
increas
peopl
rise
last
victim
decad
one soft
knife
betray
week
end
polic
will
far
cent
mon
fact
sentenc
also
went
societi
hurd
convict
offenc
polici
britain
2018
claim
result
latest
robberi
secretari
number
hope
antiprison
serious
even
remain
crimin
per
half
fail
term
british
prison
public
popul
percent


simpl
leo
referendum
countri
punish
like
can
jail
full
vote
convict
yet
time
peopl
thing
realli
know
just
crime
prison
real
think
law
world
acid
capit
useless
trianyway
must
drug
violent
attack
will
call
system
sever
lawyer
chang
stop
import
need
bestgreedi
wors
remain
corrupt
live
sentenc



The R programming code is the same shown in annex 4 titled “Comparison between article’s text and comments”

Jaccard similarity value calculated is 0.28