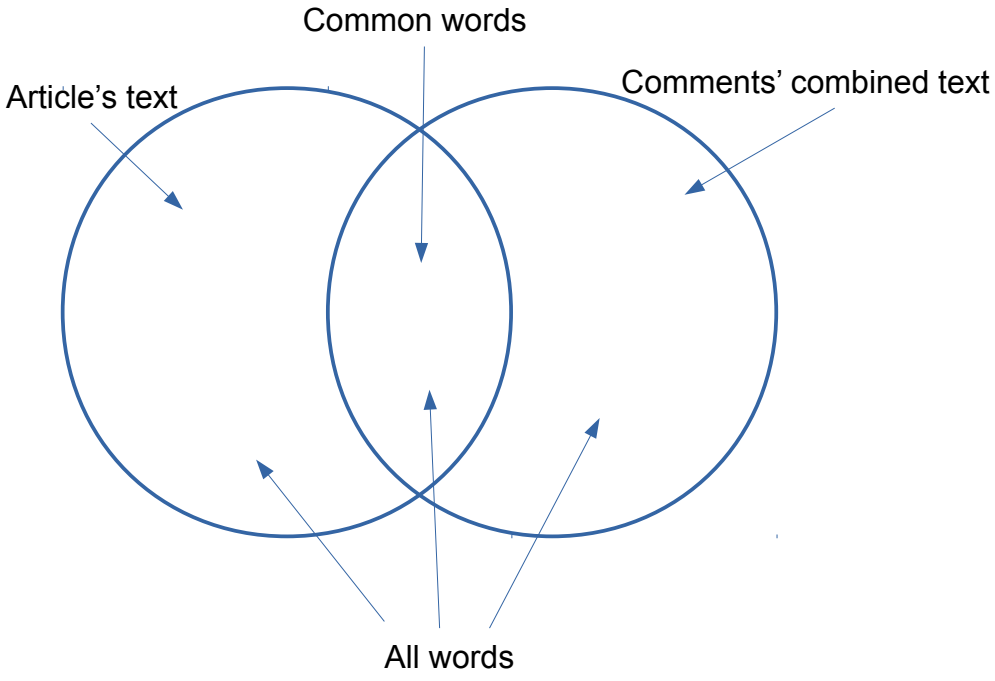
Annex 4

Comparison between article's text and comments

Various methods for text comparison exist, this dissertation's time constrains force in selecting a method that is well tested, scientifically acceptable, with a steep learning curve and, most important of all, quickly implementable. Taking in consideration that is possible to ensure, within acceptable limits, the semantic similarity because both the article's text and the set of comments about the article they all focus on the same matter. A remarkable deviation in semantics is less probable when the subject is the same. Anyway a manual analysis of the meaning of comments has already be done in the beginning of the overall analysis and it is commented in this dissertation. The text comparison of this section aims to weight the different user comments that shapes a network. The lexical similarity will be checked for surface closeness. The chosen text similarity test will be based on Jaccard similarity. It counts all the terms in common and divides the figure with the total number of term of all texts. The next illustration shows all those parts.

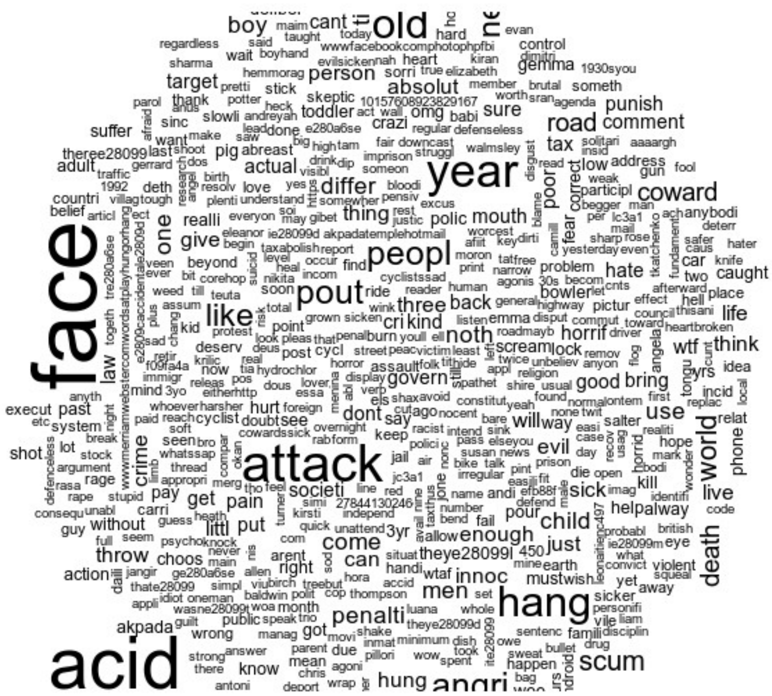


The method is similar to the one adopted by Welbers et al (2017), the same paper, citing Grimmer and Steward (2013), states that a lot of existing applications and research prove that word frequency provide with enough information for many types of analysis.
More two papers Jiang et al (2011) and Meilian et al (2014) have been source of inspiration for the Jaccard similarity research adopted in this dissertation.

The first step reduces texts to corpora, structured collection of words that allow to treat the text as a set of words in order to compute them with Jaccard similarity method.
Corpora are obtained by filtering out all unnecessary words, like for example stop words, by stemming words and by translating emoticons.

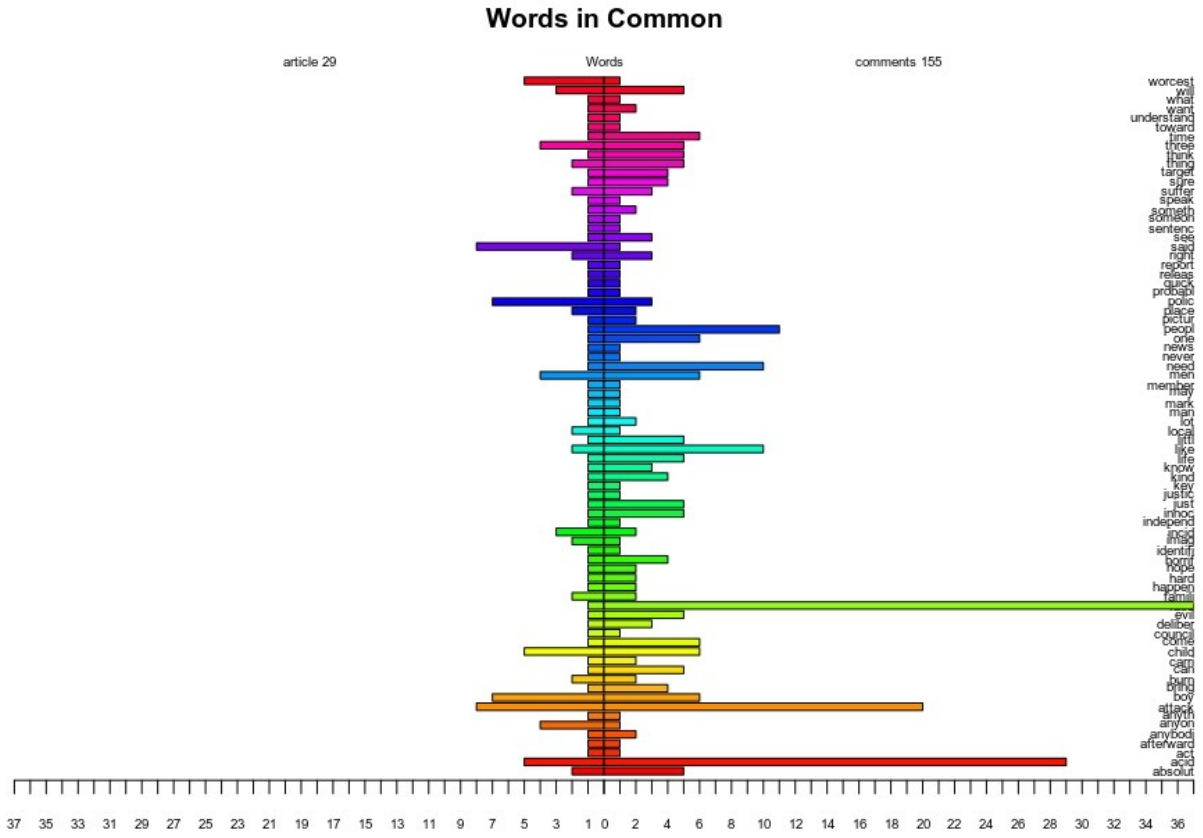
The next two illustrations are the representations of corpora from a newspaper's article and from the set of comments to the article in form of word cloud. The representation is obtained by means of R programming language.

polic
boy
said
attack
worcest
bargain
child
acid
home
inform
anyon
three
serious
hospit
mercia
sunday
investig
incid
small
men
west
substanc
store
injur
perpetr
place
retail
close
remain
shock
urg
implic
imag
discharg
supt
will
suffer
contact
longterm
absolut
ambul
hill
treat
call
concern
local
shop
burn
young
famili
thing
travi
right
citi
like
uncertain
council
worcestershopattack2jpgth



face
acid
attack
year
hang
old
peopl
pout
coward
road
death
world
scum
hung
angri
penalti
like
target
person
absolut
differ
mouth
thing
crime
law
men
innoc
come
can
child
sick
live
use
good
bring
think
evil
societi
throw
enough

The second step compares the two corpora and compute the common words. The next illustration shows the diagram obtained by extracting common words among the two corpora. The diagram is drawn by means of “pyramid” function available in R programming language.



The results of comparison is shown in the table at the next page

date of article	media	Comments media	Comment n.	article n.	Jaccard value	Comments network shape	Comments network diameter
22/07/18	Daily Star	Twitter	172	129	0	Line	1
23/07/18	Daily Star	Twitter	175	128	0	Line	1
24/07/18	Daily Star	Twitter	176	127	0	Line	1
25/07/18	The Guardian	Twitter	171	96	0	Line	2
28/07/18	The Guardian	Twitter	170	93	0	Line	1
22/07/18	Evening Standard	Twitter	216	117	0	Simple star	2
22/07/18	Independent	Twitter	194	29	0.01	Line	2
23/07/18	The Sun	Twitter	187	90	0.01	Line	1
24/07/18	Daily Express	Twitter	206	104	0.01	Line	2
23/07/18	Evening Standard	Twitter	219	116	0.01	Simple star	2
23/07/18	The Sun	Twitter	178	91	0.01	Simple star	2
22/07/18	Evening Standard	Twitter	213	111	0.01	Simple star with line	3
22/07/18	Daily Express	Twitter	207	110	0.02	Line	2
23/07/18	Metro	Twitter	203	49	0.02	Line	1
24/07/18	The Sun	Twitter	186	87	0.02	Line	1
28/07/18	Metro	Twitter	204	37	0.02	Line	1
28/08/18	The Sun	Twitter	180	80	0.02	Line	1
25/07/18	Daily Star	Twitter	173	124	0.02	Simple star	2
22/07/18	Independent	Twitter	193	29	0.03	Line	2
23/07/18	The Sun	Twitter	181	91	0.03	Line	1
22/07/18	Daily Record	Twitter	222	140	0.04	Simple star	2
25/07/18	Daily Mirror	Twitter	189	12	0.06	Simple star	2
24/07/18	The Sun	Twitter	182	87	0.07	Simple star	2
25/07/18	The Sun	Twitter	183	83	0.07	Simple star	2
25/07/18	Daily Mirror	Twitter	188	12	0.07	Simple star with line	4
22/07/18	Daily Star	Facebook	145	129	0.07	Simple star with lines	4
25/07/18	Metro	Twitter	201	47	0.08	Simple star with line	3
25/07/18	The Guardian	Twitter	167	94	0.09	Line	3
23/01/19	Daily Record	Facebook	165	136	0.1	Simple star with stars and lines	3
25/07/18	The Sun	Twitter	184	85	0.11	Line	3
23/07/18	The Sun	Twitter	179	91	0.11	Simple star with line	3
24/07/18	Daily Mirror	Facebook	150	8	0.12	Complex star with stars and lines	4
25/07/18	The Sun	Facebook	149	85	0.12	Simple star with lines	4
25/07/18	Daily Record	Twitter	223	139	0.14	Line	2
24/07/18	Metro	Facebook	157	34	0.15	Complex star with stars and lines	4
22/07/18	Metro	Twitter	199	38	0.15	Simple star	2
22/07/18	Metro	Twitter	198	38	0.15	Simple star with line	3
31/01/19	Daily Mirror	Facebook	153	22	0.17	Simple star with stars and lines	4
23/07/18	Metro	Facebook	158	49	0.18	Complex star with stars and lines	4
22/07/18	Daily Record	Facebook	162	140	0.18	Simple star with lines	4
22/01/19	Daily Record	Facebook	161	137	0.19	Simple star with lines	4
30/01/19	Independent	Facebook	228	32	0.2	Complex star with stars and lines	4
06/03/19	Independent	Facebook	230	231	0.2	Complex star with stars and lines	4
22/07/18	Daily Record	Facebook	163	131	0.21	Complex star with lines	4
23/01/19	Daily Mirror	Facebook	154	26	0.21	Simple star with stars and lines	4
22/07/18	The Guardian	Twitter	169	95	0.22	Simple star with star and line	4
07/02/19	Daily Mail	Facebook	159	53	0.25	Complex star with stars and lines	3
24/07/18	The Sun	Facebook	147	87	0.25	Simple star with lines	4
22/01/19	The Sun	Facebook	146	79	0.25	Simple star with lines	4
22/07/18	Daily Record	Facebook	164	138	0.26	Simple star with lines	4
23/01/19	Independent	Facebook	229	30	0.27	Complex star with stars and lines	4
22/07/18	Independent	Facebook	155	29	0.28	Complex star with stars and lines	4
24/07/18	The Sun	Facebook	148	227	0.32	Complex star with stars and lines	4
22/07/18	Daily Express	Facebook	160	110	0.33	Complex star with stars and lines	4

The description of the code:

Comparison is based on words that compose the texts, frequency of those words is taken in consideration for measurement.

The R libraries needed are:

```
stringi, readr, textclean, tm, data.table
```

Then proceed with the acquisition of the article's text and the set of comments about the article.

```
ar <- read_file(file_article)
fb <- read.csv(file_comment)
cm <- stri_paste(fb[,3],collapse=" ")
```

Preliminary steps include the conversion of emojis and emoticons present in comments to the correspondent sentences according to a common dictionary

```
cm <- replace_emoji(cm)
cm <- replace_emoticon(cm)
```

removal of special characters conversion to text corpus, removal of stop words (Zana, 2019), words with meaning not useful for analysis, and stemming, the removal of prefixes, infixes and suffixes from the word in order to get the word stem.

```
co_ar <- VCorpus(VectorSource(ar), readerControl = list(reader =
readPlain, language = "en"))
co_ar <- tm_map(co_ar, content_transformer(tolower))
co_ar <- tm_map(co_ar, removeWords, stopwords("english"))
co_ar <- tm_map(co_ar, stemDocument)
co_ar <- tm_map(co_ar, stripWhitespace)

co_cm <- VCorpus(VectorSource(cm), readerControl = list(reader =
readPlain, language = "en"))
co_cm <- tm_map(co_cm, content_transformer(tolower))
co_cm <- tm_map(co_cm, removeWords, stopwords("english"))
co_cm <- tm_map(co_cm, stemDocument)
co_cm <- tm_map(co_cm, stripWhitespace)
```

From text corpus construct matrix of terms for performing statistics

```
tdm_ar <- TermDocumentMatrix(co_ar)
tdm_cm <- TermDocumentMatrix(co_cm)
```

and remove sparse terms, those terms that do not influence the general meaning of the text because of their sparsity

```
tdm_ar <- removeSparseTerms(tdm_ar, 0.2)
tdm_cm <- removeSparseTerms(tdm_cm, 0.2)
```

Prepare tables for Jaccard similarity calculation (Jiang et al, 2011), (Meilian et al, 2014)

```
df_colnames <- c('word','freq')
df_ar <- as.data.frame(as.matrix(tdm_ar), make.names = FALSE)
setDT(df_ar, keep.rownames = TRUE)[]
names(df_ar) <- df_colnames
df_cm <- as.data.frame(as.matrix(tdm_cm), make.names = FALSE)
setDT(df_cm, keep.rownames = TRUE)[]
names(df_cm) <- df_colnames
df_intersect <- merge(df_ar, df_cm, by.x = "word", by.y = "word",
all = FALSE)
df_intersect_colnames <- c('word','freq_ar','freq_cm')
names(df_intersect) <- df_intersect_colnames
```

```
df_union <- merge(df_ar, df_cm, by.x = "word", by.y = "word", all
= TRUE)
df_union_colnames <- c('word','freq_ar','freq_cm')
names(df_union) <- df_union_colnames
df_union[is.na(df_union)] <- 0
```

Finally the calculation of Jaccard similarity index (Sieg, 2018), (Ma, 2018)

```
jac <- (sum(df_intersect$freq_ar) + sum(df_intersect$freq_cm)) /
(sum(df_union$freq_ar) + sum(df_union$freq_cm))
round(jac,2)
```

In order to have a better perception of the whole process is possible to visualize the word clouds

```
library(wordcloud)
wordcloud(df_ar$word, df_ar$freq, min.freq=1)
wordcloud(df_cm$word, df_cm$freq, min.freq=1)
```

and common words pyramid

```
library('plotrix')
pyramid.plot(df_merged$freq_ar, df_merged$freq_cm, labels =
df_merged$word, main = "Words in Common", laxlab = NULL, raxlab =
NULL, unit = NULL, top.labels = c(paste0('article ',article_n),
"Words", paste0('comments ',fb_comment_n)), labelcex = 0.6, gap =
0)
```

References

Sieg, A., 2018, “Text Similarities : Estimate the degree of similarity between two texts”, visited on august 2019, https://medium.com/@adriensieg/text-similarities-da019229c894

Ma, E., 2018, “3 basic Distance Measurement in Text Mining”, visited on august 2019, https://towardsdatascience.com/3-basic-distance-measurement-in-text-mining-5852becff1d7

Jiang, J., Cheng, W., Chiou, Y., Lee, S., 2011, "A similarity measure for text processing," 2011 International Conference on Machine Learning and Cybernetics, Guilin, pp. 1460-1465.

Meilian, L., Zhen, Q., Yiming, C., Zhichao, L., Mengxing, W., 2014, "Scalable news recommendation using multi-dimensional similarity and Jaccard-Kmeans clustering", Elsevier, Journal of Systems and Software, vol 95, pp 242–251

Welbers, K., Van Atteveldt, W., Benoit, K., 2017, “Text Analysis in R”, Routledge, Communication Methods and Measures, vol. 11, n. 4, pp 245–265

Zana, A.I., 2019, “List of English Stop Words”, visited on august 2019, http://xpo6.com/list-of-english-stop-words/