***Date: 28 May 2020***

**Text mining and qualitative research on transcripts of oral evidence given to the parliamentary inquiry entitled 'Social media data and real time analytics'**

CN8001 Coursework / Student No. 1720146

Chung Hoi CHIU

Abstract:

This coursework reports the following:

1. The processes and findings of using various text cleaning techniques and text mining techniques to clean and study the three transcripts of oral evidence given in the meetings held by the Science and Technology Committee of the House of Commons in 2014. The oral evidence is part of the contributions provided to the inquiry of the committee entitled 'Social media data and real time analytics'. The transcripts are hereinafter collectively referred to as 'Documents' (They and other relevant materials can be downloaded from <https://www.parliament.uk/business/committees/committees-a-z/commons-select/science-and-technology-committee/inquiries/parliament-2010/social-media-data-and-real-time-analytics/> ).

2. The results of analysing the Documents with statistical methods (a hypothesis test and four chi-squared tests) and with qualitative research methods by using the R package RQDA.

Data files, word count files required to be produced, R code used for doing this coursework and other materials can be downloaded from <https://github.com/ericchchiu/u1720146_CN8001_data> .

1. Text cleaning the Documents

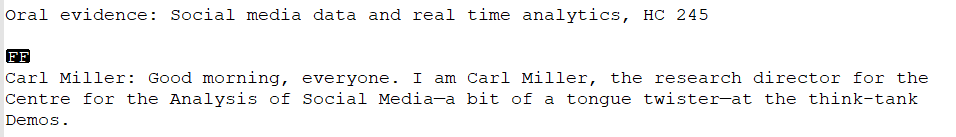
The procedures for changing the pdf format files of the Documents to plain text format files, cleaning them and adjusting their layout are as follows:

a. Open the pdf format files with Adobe Acrobat Reader DC.

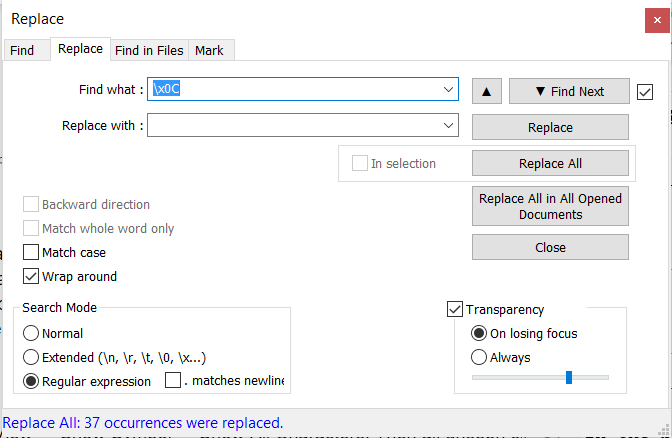
b. Choose File -> Save as Text to save each of them to a text file.

c. Open the text files with Notepad++.

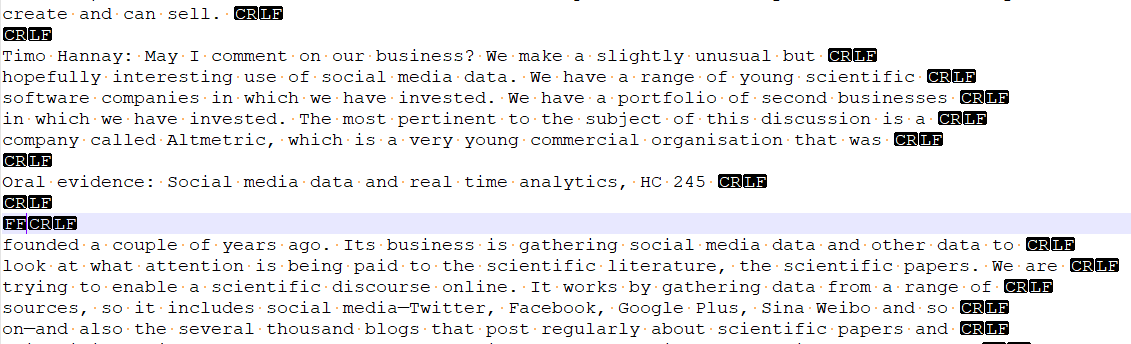
d. For each text file, observe whether there are strange characters with black backgrounds. Here is an example:



The character pair 'FF' is a 'Form Feed' symbol, ASCII 12. It informs the printer that there should be a page break. These symbols should be deleted by choosing Search -> Replace (or its corresponding icon). Then, in the opened window, select 'Regular expression' (at the bottom left corner of the below picture) and then replace all \x0C with blanks (See below picture) (\x indicates a hexadecimal figure escape and hexadecimal 0C = decimal 12).



e. Choose View -> Show Symbol -> Show All Characters. Then all unseen characters can be seen. Below is an example:

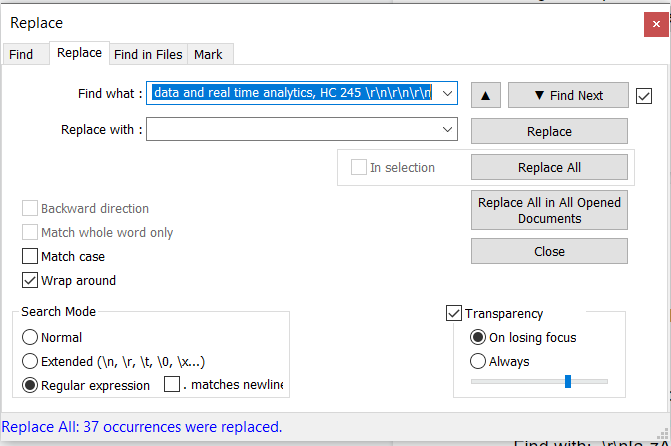


We need to delete the document name that originally appeared at the bottom of every pdf page (the 8th line from the bottom of the above picture. The FF sign on the 6th line from the bottom has already been deleted by the operation set out in d.). Then perform the following operation:

Find with: \r\nOral evidence: Social media data and real time analytics, HC 245 \r\n\r\n\r\n

Replace: [leave it blank]

A picture is shown below:



(The long line of the regular expression shown above should be close to the capacity limit of a notepad++. A longer and more complicated line may cause the notepad++ become stuck)

f. Join narratives broken in the middle:

The operation:

Find with: .\r\n[a-zA-Z0-9]

Replace All: [*one space*]

(Attention: 1) There is a dot at the beginning of the above regular expression. It means any character; and 2) replace by a space to avoid two words being joined together)

g. delete all blank lines:

The operation:

Find with: \r\n\r\n

Replace All: \r\n

(As continuous multiple blank lines may exist, the above operation should be repeated until the message 'Replace All: 0 occurrences were replaced.' appears at the bottom left corner of the window.)

h. make sure words are separated with one space only

The operation:

Find with: [*two spaces*]

Replace All: [*one space*]

(The above operation should be repeated until the message 'Replace All: 0 occurrences were replaced.' appears at the bottom left corner of the window.)

i. Since the files will be analysed with computers, all non-basic ASCII characters (that is, those characters not within the first 128 unicodes) should be converted to their basic ASCII character equivalents. The method of using notepad++ to do this is as follows:

Choose Search -> Find Characters in Range (at bottom) -> Non-ASCII Characters (128-255)

Then use the pop-up window to search for those non-basic ASCII characters. Then use the window of Search -> Replace to replace all non-basic ASCII characters with their corresponding basic ASCII characters (i.e. two pop-up windows should be used simultaneously). Examples of the conversion of non-basic ASCII characters: a) all right single quotation marks should be changed to single quotes; b) all en dashes should be changed to hyphens; and c) all pound sterling signs should be changed to GBP.

The above approach should be more convenient than using the regex [^\x0-\x7F] (representing non-basic ASCII code. Hexadecimal 0-7F is equivalent to decimal 0-127 (7 x 16 +15) and the caret ^ means not) to search for non-basic ASCII characters and then replace them.

(Optional) The following two lines of R code can be used to check whether a file contains non-basic ASCII characters, and, if the answer is yes, where they are:

Data <- readLines("Data.csv")

tools::showNonASCII(Data)

j. Question numbers, speakers' names and colons should then be added to every paragraph. Then, because the text files will be treated as csv files with | as the delimiter, the following operation should be made:

Find what: (^Q\d+)(\s)([A-Z,a-z,’]+\s\*[A-Z,a-z,']\*\s\*[A-Z,a-z,']\*\s\*)(:)(\s)

Replace with: \1|\3|

to change all expressions similar to the below two lines:

Q147 Professor van Zoonen: Yes, but an ombudsman is...

Q147 Dr d'Aquin: There is an element that relates to...

to:

Q147|Professor van Zoonen|Yes, but an ombudsman is...

Q147|Dr d'Aquin|There is an element that relates to...

(The correctness of the regular expressions shown above can be verified by copying the first two lines of text above to a new Notepad++ window and then performing the abovementioned operation to see whether the text can be converted to the second two lines)

The same changes can also be done with the R code below:

tx2 <- gsub(pattern = "(^Q\\d+)(\\s)([A-Z,a-z,’]+\\s\*[A-Z,a-z,']\*\\s\*[A-Z,a-z,']\*\\s\*)(:)(\\s)", replace = "\\1|\\3|", tx)

k. Lines within a file which are not intended to be recognised by R should be commented out with #. Lines which are intended to be crammed into one cell should be wrapped with a special character which is not originally contained in the texts, such as caret ^.

The following sample file 'csvText.csv' and R code illustrates how to import a csv file which uses | as the delimiter, # as the comment out character and adds ^ as an additional quote character:

#Text contained in the sample file 'csvText.txt':

1|^"Why shouldn't they dig the man up and have the Crowner?" said the dyer.

"It's been done many and many's the time. If there's been foul play they might find it out."^

2|"'Good morning, D'Mari,' called D'Juan.

""Good morning, D'Mari,"" called Jane."

3|^^^ is called caret and ' is called single quote.^

#Note 1) There must be at least a blank line after the last line.

#Note 2) The R functions of the two "s inside the cell at row 2 column 2 and the ^ at row 3 column 2 were escaped by doubling them.

The code for reading the csvText.txt into R, converting it to a 3 rows x 2 columns dataframe and viewing it is as follows:

csvText <- read.csv('csvText.txt', comment.char = '#', quote = '^"\'', sep = '|', header = FALSE, stringsAsFactors = FALSE)

View(csvText)

(Note: ^ is added as an additional quote character, ' is escaped by \, and # is designated as a comment character)

l. The following five clean-up files produced by following the above-mentioned techniques are kept in <https://github.com/ericchchiu/u1720146_CN8001_data> :

u1720146\_CN8001\_oral\_20140618.csv

u1720146\_CN8001\_oral\_20140623.csv

u1720146\_CN8001\_oral\_20140708.csv

u1720146\_CN8001\_Responsible\_Use\_of\_Data.csv

u1720146\_CN8001\_persons.csv

(Notes:

a) The penultimate file was converted from the committee’s report entitled *Responsible Use of Data*. Hereinafter referred to as the 'Final Report''.

b) The last file has four columns: sector, panel, person and description. In the sector column, witnesses were classified subjectively into four categories: academic, business, government and ngo, and committee members were designated 'member'. The person column includes all persons who attended the meetings at which the oral evidence was provided. The description column contains brief descriptions of all persons which were extracted from the Documents and the Final Report.

2. Text mining the Documents

Code referred to in this section is located in Appendix 2 of this coursework (code 01 to code 11). The code and word count csv files required to be produced can be downloaded from <https://github.com/ericchchiu/u1720146_CN8001_data> .

The code (code 01 to code 11) are briefly described below:

code 01: Set working directory.

packages ggplot2, ggthemes, qdap, RColorBrewer, RSQLite, textstem, tm and wordcloud will be used in this code. However, they will be imported only immediately before they are actually being used. A sample code for importing a package is shown below:

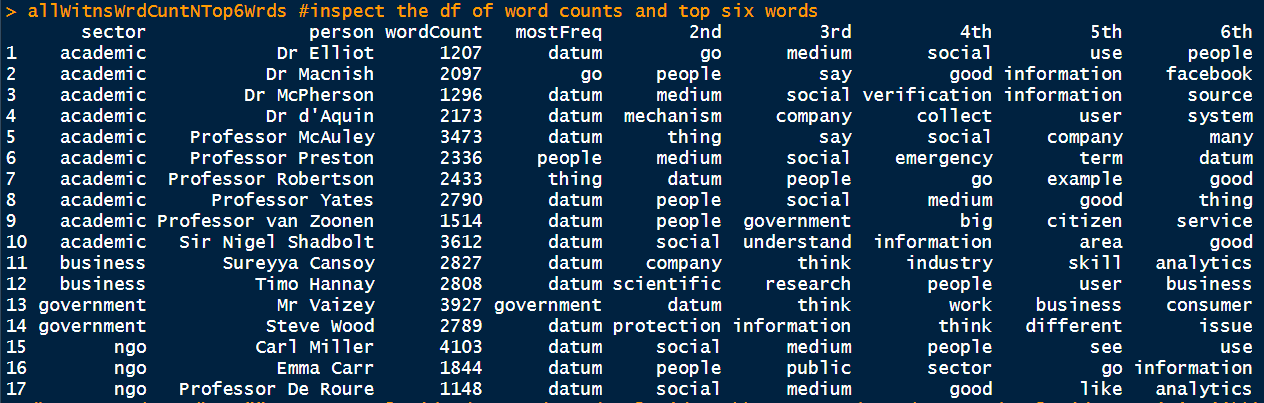
if (!require('tm')) install.packages('tm'); library(tm)

code 02: a) Import data files and b) form two dataframes.

code 03: a) Form a relational database db1 if it is not already there; b) connect R and the database; and c) copy the two dataframes created with code 02 to db1.

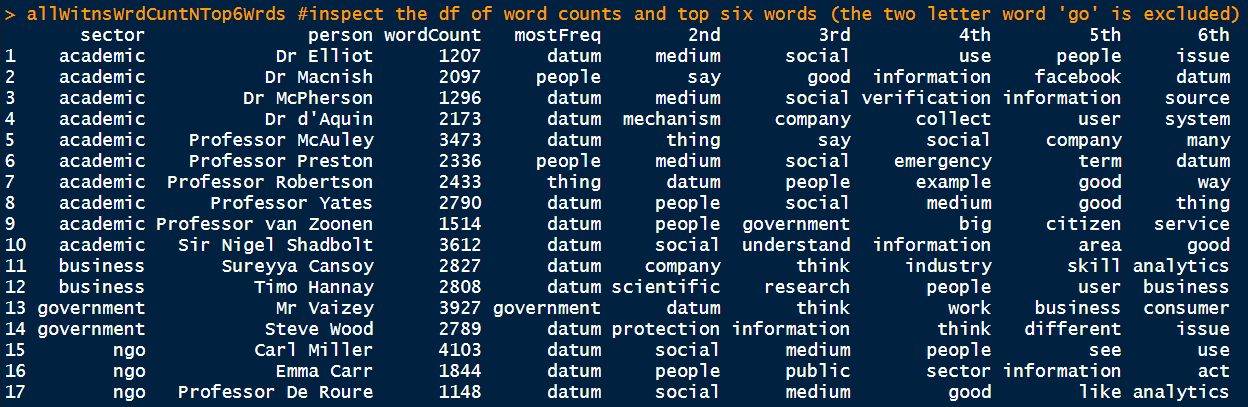
code 04: Form a dataframe of the number of words uttered by every witness and the six most frequently uttered and having substantial meaning words of every witness.

The initial output is shown below (Already superseded. For reasons please see the next paragraph):



However, it is my judgement that the word 'go' does not contain sufficient substantial meaning. Therefore, I added the control statement at.least = 3 to require the operator freq\_terms of package qdap to pick up only those words which consist of at least three characters (ignoring words having fewer than three characters is a common practice in most branches of the text mining discipline. It is even the default setting of TermDocumentMatrix of package tm).

The new dataframe, with all 'go' deleted, is shown below:



When forming the above dataframe, the operator lemmatize\_strings of package textstem was used to lemmatise words. Lemmatisation means converting all inflected forms of a word into their base form (lemmatisation will change 'am', 'are', and 'is' to 'be'; and 'writing', 'wrote' and 'written' to 'write'). Accordingly, in the above dataframe, 'datum' instead of 'data' and 'medium' instead of 'media' are shown. I regret that I still have no time to study how to use the part-of-speech tagging technique or other techniques to lemmatise the word 'saw' properly in the following three sentences: a) I saw him imbibing liquor; b) This type of blade is called a TCT saw blade; and c) You have to saw the wood into proper lengths.

code 05: Use the two dataframes created with code 02 to create a dataframe witnssUttrnces which consists of four columns: 'sector', 'person', 'question' and 'oralEvidence'.

When doing this coursework, the original data stored in csv format files was converted to dataframes and then to tables stored in the relational database db1 created with package RSQLite (code 02). When performing calculations with R, only those parts of the data that will be used to perform the calculations will be retrieved from the database db1 by using the RSQLite code. This approach has the following merits:

a. The expensive internal memory will not be occupied by unused data.

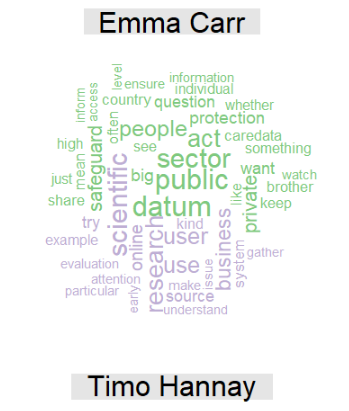
b. SQL is more versatile than R in organising, dividing and merging structured data (for example, it is simpler to use SQL to retrieve all utterances of a specified witness that were replies to questions raised by the Chair).

c. The time used to retrieve data from tables of a relational database is quicker than from a csv format file.

d. A relational database still will be there after R is exited.

However, using R's factor function to categorise dataframes is still an indispensable technique. For example, some operators of the R package e1071 accept only data categorised with factors. Therefore, in the next section of code, the code 06, factor is used instead of SQL to split a dataframe.

code 06: Form a dataframe that contains utterances of Emma Carr (the acting director of an NGO called Big Brother Watch) and Timo Hannay (the managing director of a private company called Digital Science) only and is factored into two levels: 'Emma Carr' and 'Timo Hannay'. Then, after a series of operations, a comparison cloud and a commonality cloud as shown below are produced.



A comparison cloud A commonality cloud

(Emma Carr vs. Timo Hannay) (Emma Carr and Timo Hannay)

(Please see code 04 above for the reason why datum instead of data and medium instead of media are shown in the above word clouds)

code 07: Produce functions that will be used in code 09 to perform text mining. The reasons for bundling operations into functions include: a) to avoid repetition of code; and b) to make maintenance of the code easier.

The functions are briefly described below:

7.1 is for forming a corpus and then cleaning and lemmatising it.

7.2 is for forming a word cloud with a corpus.

7.3 is for forming a tdm with a corpus (note: the default setting of TermDocumentMatrix ignores words consisting of less than three characters. Therefore, words such as go, do and up are ignored).

7.4 is for producing a csv file of word count list from a tdm (the list will be sorted first according to scores (descending) and then according to alphabetical order (ascending)).

7.5 is for producing a word count list directly from a corpus with package qdap. if the at.least control operator is set at 3, it will produce a list identical to that produced by function 7.4.

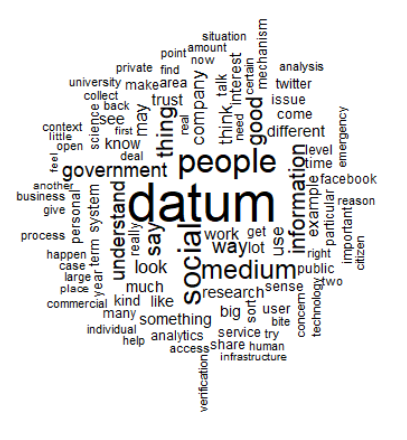
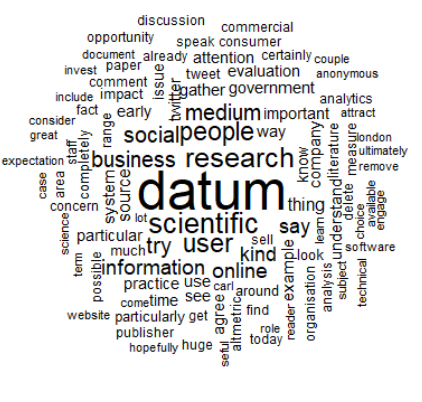
7.6 is for producing an association word list from a dtm or a tdm with the operator findAssocs of the tm package.

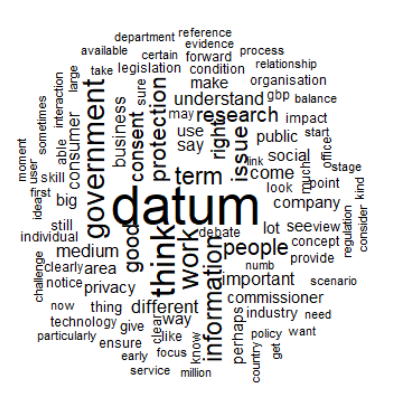
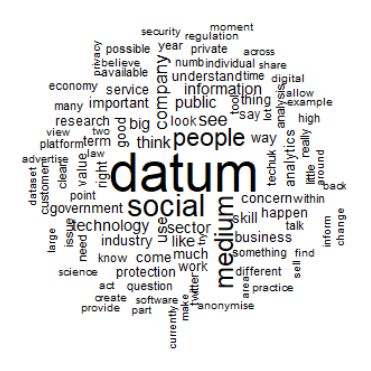
7.7 It is for converting an association word list to a three column dataframe header of which consists of three columns: 'word', 'assocsWord' and 'corrScore'.

code 08: Form four dataframes by using the SQL table witnssUttrnces created with code 05. The dataframes contain utterances of witnesses of the four sectors subjectively divided according to the information given by the Documents: academic, business, government and ngo.

code 09: Apply functions created with code 07 to the four dataframes created with code 08 to perform text mining.

code 9.2 Produce the following four word clouds with function 7.2:

  Academic sector word cloud Business sector word cloud

Government sector word cloud NGO sector word cloud

(Please see code 04 above for the reason why datum instead of data and medium instead of media are shown in the above word clouds)

code 9.4: Create the following four csv files with function 7.4: academicUttrncesWordCount.csv, businessUttrncesWordCount.csv, govtUttrncesWordCount.csv and ngoUttrncesWordCount.csv . They can be found from <https://github.com/ericchchiu/u1720146_CN8001_data> .

code 9.5: Use freq\_terms of package qdap (function 7.5) to produce four word count lists which should be identical to the lists contained in the four csv files mentioned above.

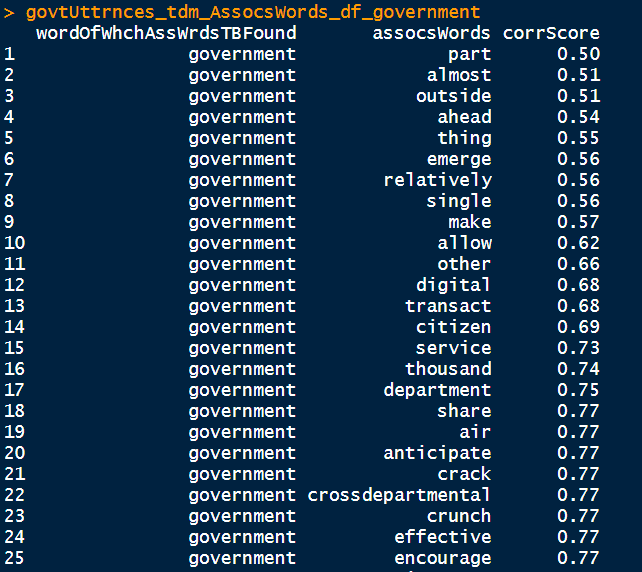
code 9.6: Bind together the four word count lists created with code 9.5 to form a dataframe. It is shown below:

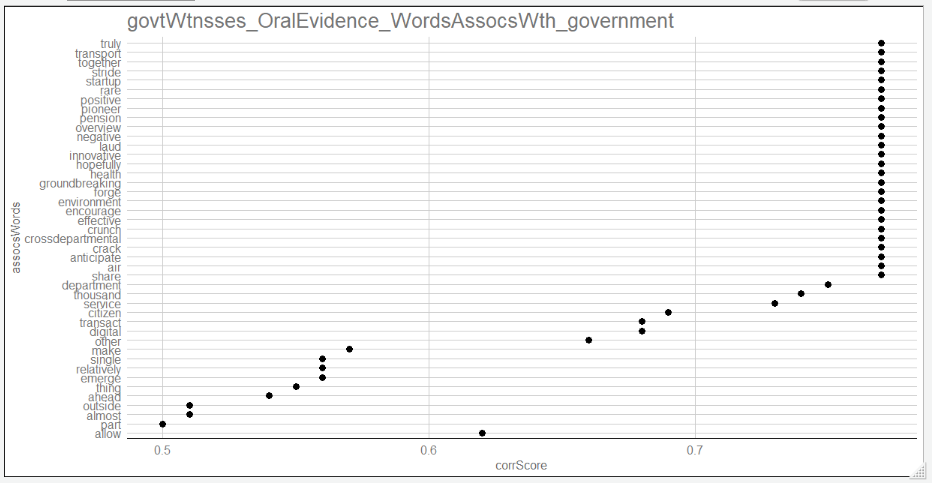


The four columns shown above should be identical to the four lists contained in the four csv files mentioned in the previous paragraph.

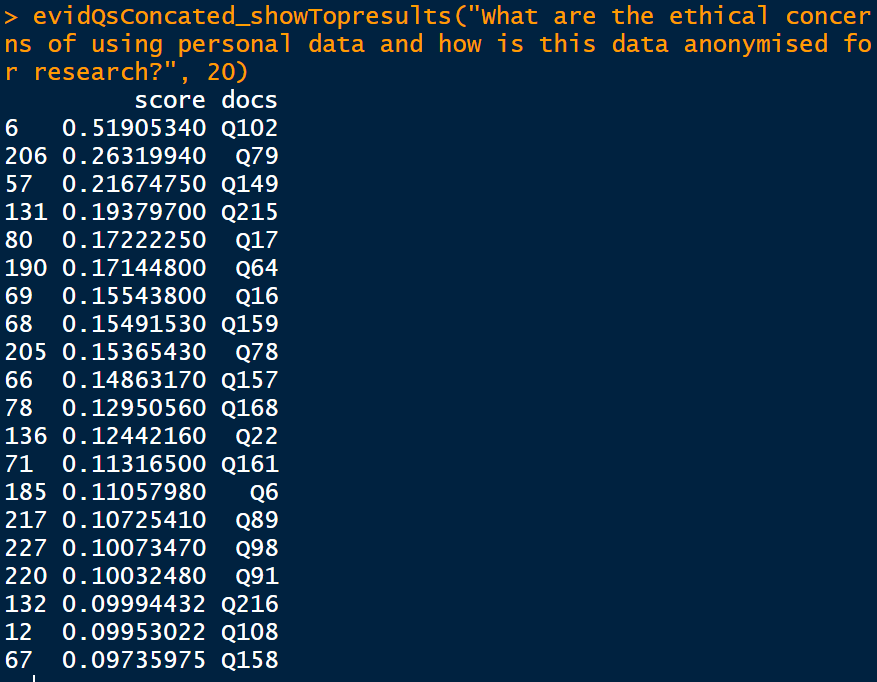
code 9.7: Use function 7.6 to find words associated with certain selected words of the four sectors. The words selected are datum, government, legislation and research. The minimum degree of association for these four words are set at 0.4, 0.5, 0.5 and 0.6 respectively.

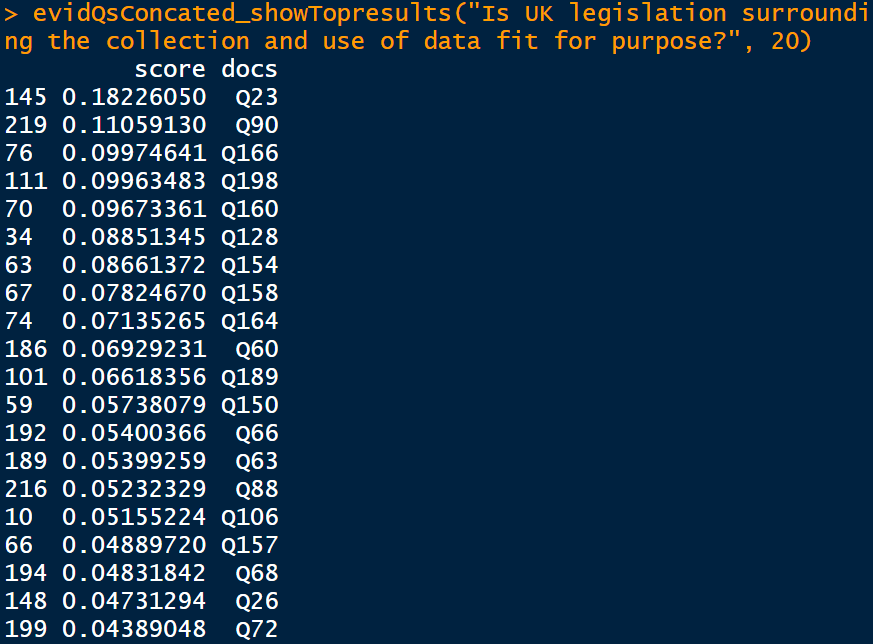
code 10: Use the list of words associated with 'government' in the government sector's utterances to form a dataframe (part of it is shown below) and then a graph:





code 11: Attempt to use tf-idf (term frequency - inverse document frequency), a weighting technique for, among other usages, ranking a set of predetermined documents according to their relevance to a set of predetermined words (Please see (Bhdat, et al., 2015) for a brief description of this technique) to find the questions and answers which are most related to the six terms of the terms of reference as contained in Section 1 of the Final Report. The lists of the top 20 most related questions to terms iv and vi of the six terms of reference are shown below. However, they are not accurate.





The failure of this code 11 is mainly because the words and phases of the terms of reference are not properly augmented by their synonyms (Bhdat, et al., 2015). However, tf-idf is an important and versatile technique and document classification is an important text mining subdiscipline. They are both worth delving into further.

3. Analysing the Documents

a. Experience gained from analysing the Documents with the Gioia method

After researching several qualitative research methods, I decided to use the Goiria method to analyse the Documents. If qualitative research methods are placed along an inductive-deductive spectrum, the Goiria method should be clearly at the inductive side. The following is a very brief summary of this method:

The Gioia approach is essentially about theory building or discovery and seeks to

generate and develop new concepts and theories.

(Candra and Shang, 2019, p. 2)

In the Gioia method, no hypotheses, themes, categories or codes should be produced in advance. They should be the products that 'emerge' from carefully planned research and painstakingly conducted perusing and coding. The research objects, such as the interviewees, should be viewed as 'knowledgeable agents' (Gioia, et al., 2012), and coding should be based on their thoughts as expressed during their interviews. Initially, the application of this method will result in hundreds of codes. For example, a PhD student generated 715 initial codes from transcripts of 23 hours of interviews ([Ref??]). The initial codes should then be carefully studied and reorganised. Then, the researcher should act as a 'knowledgeable agent' and carefully find themes and ideas that 'emerge' from the codes. This method is said to be a useful tool for showing 'gatekeepers' (editors and reviewers) (Candra and Shang, 2019, p. 37) that the findings and quotes presented in an article were not just cherry-picked (Gioia, et al., 2012). However, after coding on the Documents for more than a fortnight, I abandoned this method, because I found that I am not actually doing social science research. I should apply the deductive approach and use the six terms of the terms of reference of the inquiry and the Final Report to form an initial coding scheme. Then, I should gradually augment and improve the scheme while conducting coding and then forming my opinion on the summary of the Documents contained in the Final Report. Accordingly, I redid coding. The coding scheme is contained in Appendix 1 and the comparisons of the coding scheme to the terms of reference of the inquiry and the Final Report are contained in subsections e. and f. below respectively. Some findings about RQDA which I think have not been mentioned in literature are discussed in subsection b. below. A hypothesis test and four chi-squared tests on the Documents are discussed in subsections c., d. and e. below.

b. Some special features of RQDA

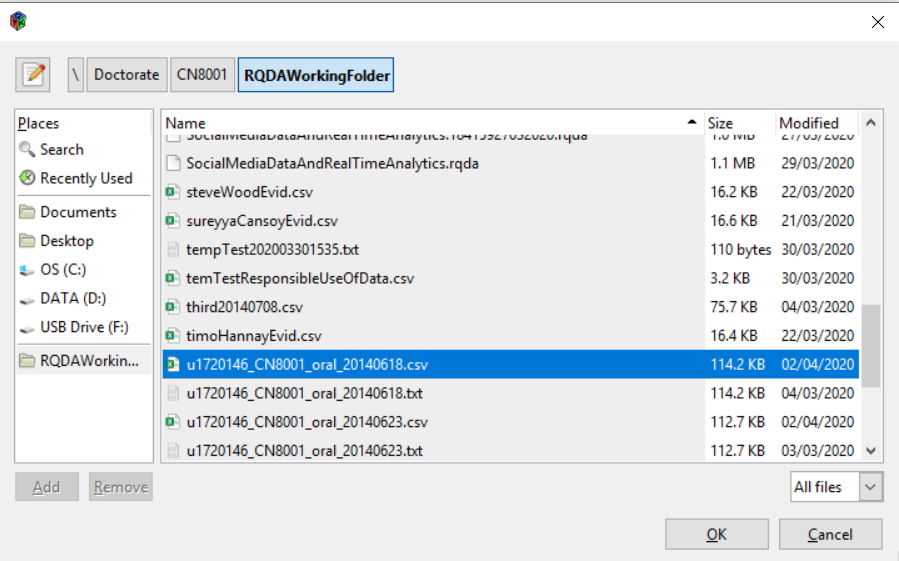
RQDA is an R package for conducting computer assisted qualitative data analysis (CAQDAS). A book published last year focuses on teaching how to use RQDA to carry out analysis (Candra and Shang, 2019). The article Estrada, 2017 also provides a detailed description of RQDA. There are also two good YouTube series which teach how to use RQDA (RQDAtuto, 2012 and the one produced by Atiq Rehman in 2018). I will hereinbelow discuss only those important features which I have not come across in literature:

i. Coding functions cannot work properly if the text is long.

A text file imported to RQDA will be crammed into a cell of an RSQLite table. Although a cell can accommodate a very long text, functions such as add/delete code, mark/unmark text, anno (annotation) text, and provide memo to codes cannot work properly if the text is long. I first joined the texts of the three files of Documents together consisting of about 52000 words and input it to RQDA for coding. I then found those functions cannot work properly. The problems disappeared when I code the three text files separately.

ii. RQDA is not file suffix sensitive.

This means one can input a file with, for example, csv as its suffix to RQDA for coding. However, if one wants to import a file which is not suffixed with txt, after the Files -> Import window was opened, she should change the box at the bottom right corner from 'text files' to 'All files'. Otherwise only those files suffixed with txt will be shown on the window (please see the picture below).



iii. RQDA does not have a word search function

A major shortcoming of RQDA is that all windows, including the text window and the code window, do not have word search functions. I found that, as the number of codes increased, I gradually forgot the codes that I created. Quite often, I had to create new codes just because I could not remember the existing codes. Therefore, I suggest that, when coding, the text should also be opened with a notepad++ (use notepad++, because it has a search function and the View -> Always on top function), and the codes should be copied to an Excel window (Excel can sort and search the codes).

vi. Exporting texts from RQDA and performing other R operations on them is not recommended.

Text in RQDA can be exported to R with code similar to the one below and other R operations can then be performed on it, such as producing a word cloud:

conndb1 <- dbConnect(RSQLite::SQLite(), '2ndRQDAWorkingFolder.rqda')

rqdaTexts <- data.frame(name = RQDAQuery("SELECT name FROM source"), text = RQDAQuery("SELECT file FROM source"))

However, I do not recommend this. It will be a disaster if RQDA or RStudio gets stuck after coding with it for a week. I recommend designating a working directory which will do RQDA coding only (or even will do RQDA coding on one file only).

c. A hypothesis test (by simulation, please see code 12 in Appendix 2)

I found from the Documents:

i. Four out of the total of 17 witnesses have the opinion that social media data should not just be regarded as a subset of big data (Sureyya Cansoy, Carl Miller, Professor Yates and Sir Nigel Shadbolt); and

ii. Three witnesses have the opposite view (Dr Elliot, Dr Macinish and Mr Vaizey).

However, in the Final Report, social media data is always treated as a subset of big data.

Therefore, I conducted the following hypothesis test to test whether the Final Report treating social media data only as a subset of big data is appropriate:

I made the following assumptions:

a) If 40% of the people have the view that social media data should not just be regarded as a set of big data, the Final Report should mention this accordingly.

b) The 17 witnesses constitute a simple random sample.

Therefore:

Null hypothesis H0: p0 = 40%

Alternative hypothesis H1: p1 < 40%

Number of sample: n = 17

However, the sample size is too small. A rule of thumb is that, for the operation of the Central Limit Theorem and hence z-score values can be used to calculate p-value, np0 and n(1-p0) each must not be less than 10 (Sternstein, 2017, p. 335). However, they are 6.8 (17 x 0.4) and 10.2 (17 x 0.6) respectively here.

Therefore, we must use the simulation techniques and code 12 was produced for doing it. The simulation techniques will be used if the conditions of a test procedure cannot be met, or there is no such procedure (Sternstein, 2017, p. 344).

In code 12, 100 samples of n = 17 (the total number of witnesses) are extracted from a large sample population (2000) where 40% of them agree that social media should not just be regarded as a sub-branch of big data (H0) and the procedure for finding the p-value is repeated 100 times. The p-value is then found to be about 0.125.

With this large p-value (0.125 > 0.1, one of the usual significant levels (alpha). The other two usual significant levels are 0.05 and 0.01. The smaller the value of the significant level, the more difficult to reject the null hypothesis. Therefore, for example, in new drug test, for play it safe, the null hypothesis is usually stated as that the new drug is not better than the current drug.), there is no sufficient evidence to reject H0, that is, there is no sufficient evidence to reject the belief that the true percentage of people supporting the view that social media should not just be regarded as a sub-branch of big data is equal to 40%.

I adjusted the ratio of the sample population and reran the code several times. Then I found p-value is still at about 0.055 (still larger than the significant level if it is set at 0.05) even if H0 is set at 45%.

Therefore, the Final Report does not mention the view that social media data should not just be regarded as a subset of big data is not appropriate.

d. A chi-squared test (by simulation and by fisher method, please see code 13 in Appendix 2)

I found the attitudes of the witnesses towards the EU Data Protection Legislation were quite diverse. Their sectors and attitudes are shown below:

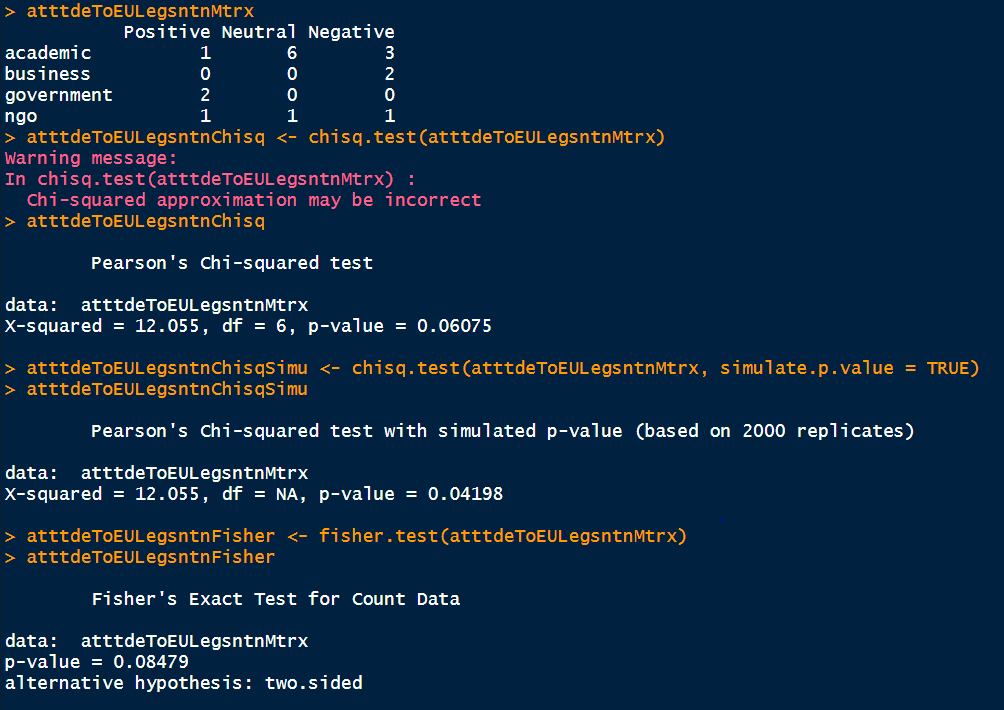
Sureyya Cansoy(business, -ve), Timo Hannay(business, -ve), Carl Miller(ngo, -ve), Professor Yates(academic, nutual), Dr McPherson(academic, nutual), Professor Preston(academic, -ve), Sir Nigel Shadbolt(academic, neutral), Professor McAuley(academic, +ve), Professor De Roure(ngo, neutral), Professor van Zoonen(academic, neutral), Dr d'Aquin(academic, -ve), Professor Robertson(academic, -ve), Emma Carr(neutral, +ve), Dr Macnish(academic, neutral), Dr Elliot(academic, neutral), Steve Wood(government, +ve), Mr Vaizey(government, +ve)

Therefore, code 13 was produced to test whether a witness's attitude towards the legislation was independent of the sector to which she belonged. The hypothesis statements are as follows:

Null hypothesis H0: a witness's attitude and the sector to which she belonged were independent.

Alternative hypothesis H1: a witness's attitude and the sector to which she belonged were not independent.

The following picture shows the results of running the code:



Explanations:

1st command: showing the table of attitudes.

2nd command: performing the chi-squared test on the table.

The warning message: It will appear whenever there is a cell the number in which is less than 5.

3rd command: showing the chi-squared test results.

df means degree of freedom. In the chi-squared test, df is equal to (row number -1) x (column number - 1). In a chi-squared test, the sums of rows and the sums of columns should be calculated first. Therefore, with row sums and column sums known, in this 4 rows x 3 columns table, only values in six out of the total 12 numbers can be 'freely' attributed. so df = 6.

4th command: performing the simulation chi-squared test on the table.

Since the traditional chi-squared test may give an incorrect result, the simulation method was used. Code 12 above provided a simple simulation example, but the simulation procedure here is more complicated.

5th command: showing the simulation chi-squared test result.

6th command: performing the Fisher test on the table.

7th command: showing the Fisher test result.

The p-value obtained from the simulation chi-squared test is 0.04198 while that obtained from the Fisher test is 0.08479.

If significant level is set at 0.1, both p-values 0.04198 and 0.08479 < 0.1, with such small p-values, there is sufficient evidence to reject H0 and accept H1, that is, a witness's attitude and the sector to which she belonged were not independent.

If the significant level is set at 0.05, 0.04198 < 0.05 while 0.08479 > 0.05. Therefore:

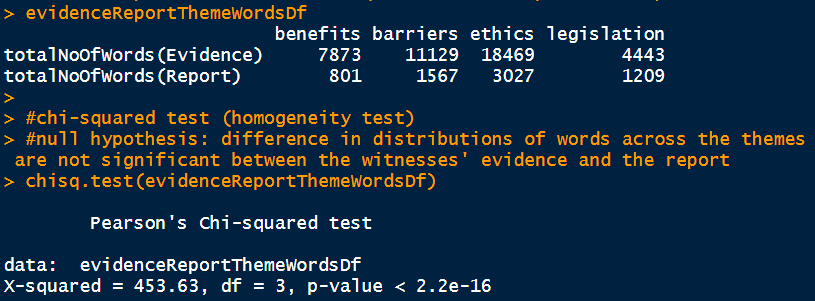
if the simulation chi-squared test result is adopted (0.04198 < 0.05), the conclusion will be the same as the above paragraph; while,

if the Fisher method is adopted, since the p-value is larger than the significant level (0.08479 > 0.05), there is no sufficient evidence to reject H0, that is, a witness's attitude and the sector she belonged to were independent.

e. Three normal chi-squared tests (please see code 14 in Appendix 2)

I found the evidence contained in the Documents and the summary of the evidence contained in the Final Report can be attributed to four themes, namely, benefits, barriers, ethics and legislation (please see the coding scheme contained in Appendix 1). One of the three following chi-squared tests is to test homogeneity and the other two are to test independence. All three tests are for studying the distribution of the words across the four themes.

i. test for homogeneity



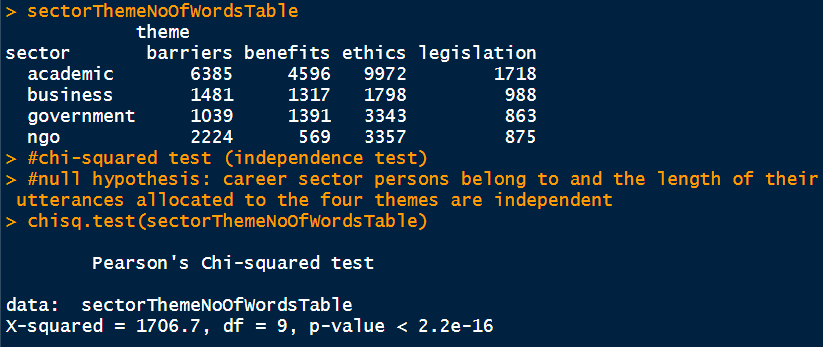
Explanations:

1st command: showing the table.

2nd command: performing the chi-squared test. The null hypothesis H0: difference in distributions of words across the four themes are not significant between the witnesses' evidence and the Final Report.

With the p-value as small as less than 2.2e-16, there is sufficient evidence to reject H0, which means that the difference in the distributions of words across the four themes are significant between the witnesses' evidence and the Final Report.

ii. test for independency



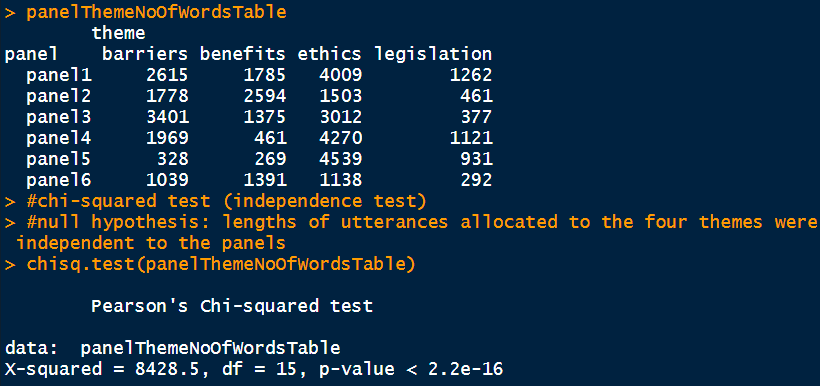
Explanations:

1st command: showing the table.

2nd command: performing the chi-squared test. The null hypothesis H0: career sector persons belong to and the length of their utterances allocated to the four themes are independent.

With the p-value as small as less than 2.2e-16, there is sufficient evidence to reject H0, which means that career sector persons belong to and the length of their utterance allocated to the four themes are not independent.

iii. test for independency



Explanations:

1st command: showing the table.

2nd command: performing the chi-squared test. The null hypothesis H0: lengths of utterances allocated to the four themes were independent to the panels.

With the p-value as small as less than 2.2e-16, there is sufficient evidence to reject H0, which means that the lengths of utterances allocated to the four themes were not independent to the panels.

f. Comparisons of the coding scheme to the terms of reference of the inquiry and the Final Report

The coding scheme is contained in Appendix 1.

The following are the salient findings after coding the Documents and the Final Report:

i. No drifting away from the terms of reference.

The evidence given by the witnesses and recorded in the Documents and the summary of the evidence contained in the Final Report are all related to the terms of reference. No drifting away from the terms of reference has been found.

ii. Term 2 of the terms of reference was not touched at all either during the evidence giving meetings or in the Final Report.

For reasons unknown, no evidence given and no passage of the summary of the Final Report are related to term 2 of the terms of reference. (It is reproduced here for ease of reference: 'How does the UK compare to other EU countries in funding for real-time big data research?')

iii. Matters concerning anonymisation were discussed in depth during the evidence giving meetings but were not mentioned at all in the Final Report.

44% of the evidence and 46% of the summary (please see the picture shown in subsection 3.e.i.) contained in the Final Report concern term 4 of the terms of reference (It is reproduced here for ease of reference: 'What are the ethical concerns of using personal data and how is this data anonymised for research?'), that is, concern the ethics theme. Anonymisation, one of the seven sub-themes (categories) of the ethics theme, is the only category specifically mentioned in term 4. Witnesses also provided a significant number of insightful and professional opinions on this category. For example, they compared the degree of difficulty of deanomysation and hacking; they pointed out that pseudonymisation, de-identification and anonymisation are different; and they provided opinions on whether anonymised personal data should still be viewed as personal data. However, very surprisingly, anonymisation was not mentioned at all in the summary of the Final Report.

iv. The opinions of the witnesses on whether social media data should be classified as a subset of big data or as a kind of data independent of big data are very diverse. However, the Final Report states clearly and conspicuously in the first paragraph of its summary that social media data is 'a subset of big data' and no mention was made of the opposite opinions.

v. In addition to the omissions mentioned above, the witnesses gave a significant number of other important information and opinions that are not mentioned in the Final Report, such as the usual black box nature of social media data handling software tools and the probabilistic nature of social media data.

4. Conclusion

My opinions on the tools used for doing this coursework are as follows:

a. Notepad++: It is a versatile and convenient tool for text cleaning and text editing.

b. RQDA: The major shortcoming of this CAQDAS tool is that it has no word search function. It is akin to prohibiting flipping a book's pages while reading it (please see subsection 3.b.iii.). Another shortcoming of this tool is that coding functions always cannot function properly, if the text is too long (please see subsection 3.b. i. 52,000 words is considered too long). In addition, one working directory should be used to code one document only, if the document is relatively long, and the working directory should not be used to perform other R tasks (please see subsection 3.b.iv.). The RQDA was stuck several times during the three weeks that I used it to code. According to my diary, on two or three occasions, to reinstate RQDA, I needed to go to the following directory to add something to it (perhaps the errors were due to plotting functions of RQDA):

C:/Users/Eric/AppData/Local/Temp/

c. R Language: The first sentence of the official website of R (<https://www.r-project.org/>) is 'R is a free software environment for statistical computing and graphics'. This means that R is not a general-purpose programming language. Therefore, when learning R, one should focus on learning which package and which of its functions is most suitable for performing which statistical or machine learning tasks. Learning design patterns (formalised good programming practices) is less important, although it is very important in learning general-purpose programming languages.

My opinions on the Documents and the Final Report can be found in subsection 3.f. To avoid giving the impression that the summary of a document is a cherry-picking process, perhaps one should adopt the Gioia method (Gioia, et al., 2012; and Candra and Shang, 2019. Please also see subsection 3.a.) or other similar qualitative research methods to do the summary.

References

Bhdat, D. et al. (2015) *Text mining for central banks*. London: Bank of England.

Chandra, Y. and Shang, L. (2019) *Qualitative research using R: a systemic approach*. Singapore: Springer.

Estrada, S. (2017) 'Qualitative analysis using R: a free analytic tool', *The Qualitative Report*, 22(4), pp. 956-968.

Gioia, D. A. et al. (2013) 'Seeking qualitative rigor in inductive research: notes on the gioia methodology', *Organizational Research Methods*, 16(1), pp. 15-31.

Sternstein, M. (2017) *Barron's AP Statistics*. 9th edn. New York: Barron's.

Wishart, M. (2018) *Moral identity work in senior business managers*. Unpublished PhD thesis, The Open University, Available at: https://ethos.bl.uk/OrderDetails.do?did=2&uin=uk.bl.ethos.757660 (Accessed: 22 May 2020).

Appendix 1

The coding scheme:

|  |  |  |
| --- | --- | --- |
| Theme | Category | Code |
| Benefits  (Term 1 of the terms of reference query 1 and Section 2 of the Final Report) | For the economy | new raw material of 21stC like oil/ redraw business models/ revolutionise established disciplines/ deepen understanding of the market, consumers.../ UK is the leader of this sector/ tech companies provide tools, software, or end-to-end services\* |
| For the administration | can compensate the deficiency of the census/ addition to but not replacement of other method of inquiry/ example: quick twitter validate cctv/ census is structured and detailed while smd is not/ not all participate in sm but it is quick/ sm can be self-correcting (people would dispatch and judge smd as experts)\*/ sm can be used to monitor spread of infections\*/ UK is a leader in open data. It should also have the potential to be the leader of using smd\*/ sm is plenty and fast\* |
| Barriers  (Term 3 of the terms of reference and Section 3 of the Final Report (part)) | Reliability | methods are still young and weak/ deliberate manipulation/ black box (arcane and complex maths or even no explanation at all)\*/ cherry-picking of study results\*/ difficult to disprove cherry-picking accusations: disclosure of smd used is restrained\*/ need to triangulate data\*/ need to invent reliable verification systems\*/ whether sufficient sm physical infrastructure located in UK not very important: cloud-based/ sm infrastructure can mean software, technologies and data\* |
| UK weakness comparing with EU countries | UK emphasises govt's collection of data is for national security -> support fading gradully, suspicion growing gradully/ EU emphasises govt's collection of data is for improving services -> support and trust would not/ key questions about govt's collection of data: what they will be used for and how can I know whether their use is against my interests/ people will support the govt to use their data if they know it is for public's benefit/ govt usually cannot communicate to the public the benefit derived from using their data/ people usually hold the govt a higher standard than private companies when data protection is concerned/ private companies will provide benefits in return for the data they collected/ people are keen on being given the right of opt-out when they are providing data |
| Skills | govt's use of SMD inconsistent across the UK/ govt's sucess depends on interest levels of officials/ civil servents should have the right 'mix of skills'/ those possess the right skills prefer to work in private sector such as start-ups (govt need to pay 20% more)/ reskilling/ number of job opportunities (i.e. skilled personnel shortage) is high and its growing rate is also high/ govt should be the strong leader in overcoming the barrier of acute skilled personnel shortage/ still in infancy stage, need more research and development/ whether smd is a distinct discipline and whether it should be viewed as a part of big data\*/ requires a hybrid skills set (social science + computational science)\*/ smd is unstructured and multi-modes (photos, videos, mp3, blogs, vlogs, etc.)\*/ not just skill shortage, but also understanding shortage (sm's power and limits)\*/ resource is already there, need skills and technologies to exploit it (data capability)\*/ change immigration policies to attract people having the right skills\* |
| Ethics  (Term 4 of the terms of reference and Section 4 of the Final Report (part)) | Communication of intentions | communication of how collected data may be used was usually not clear and helpful/ govt support improve in transparency of data use/ create a set of data principles (not yet done by the industry)\*/ 60% to 70% of smd is provided to the public by twitter (much more difficult to obtain data from other sm platforms)\*/ third party use smd under licence -> so need to agree to be constrained by regulations\*/ ethics standards should be high: ask 'should i do', not just 'can i do'\*/ use fewer data and anonymising it to a deep degree\*/ dragnet approach whereby all smd is collected and trawled is not a proper approach\* |
| Platforms are in foreign jurisdictions | global nature of the internet cause confusion when considering regulation/ large international companies not based in the UK have the best ability to misuse personal data/ the EU draft GDPR extend the scope of data protection outside the EU/ should not reject competition from foreign companies\* |
| Informed consent | smd is an important asset but its use is limited by law: Data Protection Act 1998 requiring consent/ how data given will be used is not known when consent is given (whether it is an informed consent?)/ companies prefer to have clear ethic guidelines for compiling with/ signing terms and conditions does not necessarily constitue an informed consent/ people are compelled to accept terms and conditions or otherwise access to services would be denied/ people treat twitter a pub and think conversations made in twitter does not belong to twitter/ people do not have sufficient awareness that their smd is being taken up\*/ smd may be passed to an unknown third party in future (not future-proof)\*/ very sophisticated use or interpretation of smd\*/ plain English\*/ people will trust the govt more if they know data will not be kept by a central database and can circulate among departments\*/ an ombudsman\*/ entering information into an sm is not the same as broadcasting\*/ consent has its place but it is all about balance (right protections vs research needs)\* / explicit consent can lead to excessive bureaucracy and damage consumers' experience\* |
| Length and complexity of the terms and conditions | few people reading terms and conditions before signing it. it usually is long and complicated/ a slimmed-down, jargon free guide which summarising the full terms and conditions/ lengthy and difficult terms and conditions come from a culture which rely heavily on lawyers/ some organisations sometimes are able to exploit the opacity of the terms and conditions/ use icons or graphs\* |
| Require or request data | seldom provide justification for collecting data/ usually data is required for providing or providing better services/ duplicitous requirement of data (may be useful in future)/ smd will become more and more valuable since more and more advanced methods for exploiting it are invented\*/ an increasing movement of people ceasing to use sm: not because they are vulnerable but because they don't want to be tracable\*/ a misconception: once the data is public it ceases to be personal data\*/ function creep: initial intention would be morphed slowly\* |
| Kitemarks | the kitemark approach is a solution to the lack of international governmental agreement on data protection/ a kitemark provide users with confidence to a set of terms and conditions/ the crystal mark awarded by the Plain English Campaign may be a form of kitemark |
| Anonymisation | anonymised data no longer subject to data protection rules and constraints\*/ machine de-anonymisation or re-identification should be prohibited\*/ cyber-attack and hacking are more efficient than de-anonymisation\*/ rules, guidelines or standards to deal with matters related to anonymisation of smd is required\*/ pseudonymisation, de-identification and anonymisation\* |
| Legislation  (Terms 5 and 6 of the Terms of reference and part of Section 3 and part of Section 4 of the Final Report) | Rules, regulations and legislations | EU's draft GDPR is welcome: the fear of the citizen will lose out to big business interests is needed to be ractified/ EU's draft GDPR would lead to lose of millions of jobs: it would restrict direct marketing and the ability to assess credit risk/ basic data protection principles (guidance and advice) sufficient and flexible. new legislation not necessary/ a balance should be struke between regulations and the needs of data intensive research/ tension between innovation and regulations/ the right to be forgotten (the right to be forgiven)\*/ tension between right of forgotten vs freedom of expression\*/ tension between security considerations and privacy considerations\*/ current UK legislation is in general satisfied\*/ regulations will not suffocate the development of sm. It grows so rapidly\*/ companies prefer to have clear ethics standards |

Appendix 2

The code

#R code for text mining of the three transcripts of oral evidence

#given to the parliamentary inquiry

#code 01 - code 11:'Social media data and real time analytics' (HC245)

#Word counts, word associations, word clouds and

#tf-idf text analysis

#code 12 finding the p-value by simulation

#code 13 chi-squared test (independence test, by simulation and by the fisher method)

#code 14 chi-squared test (one homogeneity test and two independence tests)

#----------------------------------------

#code 01:

#set working directory

setwd(dirname(file.choose()))

getwd()

#----------------------------------------

#code 02:

#import the three oral evidence data files and

#convert them to dataframes

#then combine the three dataframes into one

#then add an index column(id) to the combined file

#then import u1720146\_CN8001\_persons.csv and convert it to a dataframe

oral0618 <- read.csv('u1720146\_CN8001\_oral\_20140618.csv', comment.char = '#', sep = '|', stringsAsFactors = FALSE)

oral0623 <- read.csv('u1720146\_CN8001\_oral\_20140623.csv', comment.char = '#', sep = '|', stringsAsFactors = FALSE)

oral0708 <- read.csv('u1720146\_CN8001\_oral\_20140708.csv', comment.char = '#', sep = '|', stringsAsFactors = FALSE)

oralAll <- rbind(oral0618, oral0623, oral0708)

oralAll$id <- seq.int(nrow(oralAll))

persons <- read.csv('u1720146\_CN8001\_persons.csv', comment.char = '#', sep = '|', stringsAsFactors = FALSE)

#----------------------------------------

#code 03:

#create a sqlite database db1 or connect to it

#copy oralAll and persons to db1 if they are not already there

if (!require('RSQLite')) install.packages('RSQLite'); library('RSQLite')

conndb1 <- dbConnect(RSQLite::SQLite(), 'db1')

dbListTables(conndb1) #inspect db1

#if tables oralAll or persons are not in db1

#create them with the below two lines of code

dbWriteTable(conndb1, 'oralAll', oralAll, overwrite = TRUE) #if db1 does not have oralALL

dbWriteTable(conndb1, 'persons', persons, overwrite = TRUE) #if db1 does not have persons

#----------------------------------------

#code 04:

#form a df of word count: top six most frequently used words of every witness

#check whether tables oralALL and persons is already in db1

dbListTables(conndb1)

#if tables oralAll and persons are already in db1

#(if not, use above code 03 to produce them)

#form a dataframe of utterances uttered by all witnesses

#ranked first by sectors then by witnesses (by alphabetical order)

#evidence uttered by each witness are concatened to form a document

evidWitnssConcated <- dbGetQuery(conndb1, "SELECT sector, persons.person AS person, GROUP\_CONCAT(oralEvidence, ' ') AS oralEvidence FROM oralALL JOIN persons ON oralALL.person = persons.person WHERE sector <> 'member' GROUP BY persons.person ORDER BY sector, persons.person, oralEvidence")

View(evidWitnssConcated) #inspect evidWitnssConcated

#use qdap to form a df of word counts for all witnesses

if (!require('qdap')) install.packages('qdap'); library(qdap)

wordCount <- word\_count(evidWitnssConcated$oralEvidence)

evidWitnssConcated$wordCount <- wordCount

evidWitnssConcated\_wordCount <- evidWitnssConcated[,c(1,2,4)]

View(evidWitnssConcated\_wordCount) #inspect the wordCount df

#a function for cleaning and lemmatising texts

#tm is requiredd for forming a corpus and cleaning

if (!require('tm')) install.packages('tm'); library(tm)

#textstem is required for lemmatising

if (!require('textstem')) install.packages('textstem'); library(textstem)

getFrmPersUttrnces\_cleanNLemmatsdCrpus <- function(evidWitnssConcated\_cell){

corpus <- Corpus(VectorSource(evidWitnssConcated\_cell))

corpus <- tm\_map(corpus, tolower)

stop.word <- unlist(read.table("stop\_word.txt", stringsAsFactors=FALSE))

corpus <- tm\_map(corpus, removeWords, stop.word)

corpus <- tm\_map(corpus, removeNumbers)

corpus <- tm\_map(corpus, removeWords, stopwords())

corpus <- tm\_map(corpus, removePunctuation)

stop.char <- unlist(read.table("stop\_char.txt", stringsAsFactors=FALSE))

corpus <- tm\_map(corpus, removeWords, stop.char)

corpus <- tm\_map(corpus, lemmatize\_strings)

corpus <- tm\_map(corpus, stripWhitespace)

return(corpus)

}

#a function for picking up the top six most frequently used lexicon words

#qdap's freq\_terms is used

top6Words <- function(x) return(freq\_terms(getFrmPersUttrnces\_cleanNLemmatsdCrpus(x), top = 6, at.least = 3, extend= FALSE)$WORD)

#above: found go a high frequency word but has little solid meaning

#so at.least = 3

#form list of lists of six words for all witnesses

listwitnss6TopWrds <- lapply(evidWitnssConcated$oralEvidence, top6Words)

#convert the list of lists to df

top6WordsDf <- as.data.frame(t(matrix(unlist(listwitnss6TopWrds), nrow=length(unlist(listwitnss6TopWrds[1])))))

#add column names

colnames(top6WordsDf) <- c('mostFreq', '2nd', '3rd', '4th', '5th', '6th')

#combine the wordCount df and the top six words df

allWitnsWrdCuntNTop6Wrds <- cbind(evidWitnssConcated\_wordCount, top6WordsDf)

allWitnsWrdCuntNTop6Wrds #inspect the df of word counts and top six words

#----------------------------------------

#code 05:

#use sqlite code to create a sql table and a dataframe of witnesses and #the utterances of them ranked by sectors, names and then

#question numbers

witnssUttrnces <- dbGetQuery(conndb1, 'WITH witnssUttrnces AS (SELECT sector, persons.person AS person, id, question, oralEvidence FROM oralAll INNER JOIN persons ON oralAll.person = persons.person WHERE persons.sector <> "member" ORDER BY sector, person, id) SELECT sector, person, question, oralEvidence FROM witnssUttrnces')

dbWriteTable(conndb1, 'witnssUttrnces', witnssUttrnces, overwrite = TRUE)

#----------------------------------------

#code 06:

#create a dataframe of Timo Hannay and Emma Carr's utterances

#use factor

THAndECUttrnces\_wthFactor <- dbGetQuery(conndb1, 'SELECT person, oralEvidence FROM witnssUttrnces WHERE person IN ("Emma Carr", "Timo Hannay")')

#if do not use factor and concatenate cell contents with sql, the above sql code should be changed to as follows:

#THAndECUttrnces <- dbGetQuery(conndb1, 'SELECT person, GROUP\_CONCAT(oralEvidence," ") AS oralEvidence FROM witnssUttrnces WHERE person IN ("Emma Carr", "Timo Hannay") GROUP BY person ORDER BY person')

#designate $person a factor

THAndECUttrnces\_wthFactor$person <- factor(THAndECUttrnces\_wthFactor$person)

#subset into two dataframes

Emma\_Carr\_df <- subset(THAndECUttrnces\_wthFactor, person == "Emma Carr")

Timo\_Hannay\_df <- subset(THAndECUttrnces\_wthFactor, person == "Timo Hannay")

#combine, aggregate and concatenate the above two dataframes

THAndECUttrnces <- rbind(Emma\_Carr\_df, Timo\_Hannay\_df)

THAndECUttrnces <- aggregate(oralEvidence ~ person, THAndECUttrnces, paste, collapse = ' ')

#clear data and produce tdm for Emma Carr and Timo Hannay

#packages required: tm(corpus and cleaning) and textstem(lemmatize\_strings)

THAndECUttrnces\_cor\_cl <- Corpus(VectorSource(THAndECUttrnces$oralEvidence))

THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, tolower)

stop.word <- unlist(read.table("stop\_word.txt", stringsAsFactors=FALSE))

THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, removeWords, stop.word)

THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, removeNumbers)

THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, removeWords, stopwords())

THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, removePunctuation)

stop.char <- unlist(read.table("stop\_char.txt", stringsAsFactors=FALSE))

THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, removeWords, stop.char)

THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, stripWhitespace)

THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, lemmatize\_strings)

THAndECUttrnces\_tdm\_cl <- TermDocumentMatrix(THAndECUttrnces\_cor\_cl)

colnames(THAndECUttrnces\_tdm\_cl) <- c("Emma Carr", "Timo Hannay")

THAndECUttrnces\_tdm\_cl

# coerce as a matrix

THAndECUttrnces\_tdm\_cl <- as.matrix(THAndECUttrnces\_tdm\_cl)

colnames(THAndECUttrnces\_tdm\_cl) <- c("Emma Carr", "Timo Hannay")

# assign a palette

# package RColorBrewer is required

if (!require('RColorBrewer')) install.packages('RColorBrewer'); library(RColorBrewer)

pal <- brewer.pal(5, "Accent")

# plot wordclouds

#package wordcloud is required

if (!require('wordcloud')) install.packages('wordcloud'); library(wordcloud)

set.seed(12345)

comparison.cloud(THAndECUttrnces\_tdm\_cl, scale=c(2,0.5), max.words = 50, rot.per = 0.3, random.order=FALSE, color=pal, title.size=2)

set.seed(12345)

commonality.cloud(THAndECUttrnces\_tdm\_cl, scale=c(5,0.5), max.words = 50, rot.per = 0.3, random.order=FALSE, color=pal, title.size=2)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

##not use factor. commented out. just for reference.

##commonality and comparison clouds for utterances of

##Emma Carr and Timo Hannay

##create a dataframe of Timo Hannay and Emma Carr's utterances(concated)

#THAndECUttrnces <- dbGetQuery(conndb1, 'SELECT person, GROUP\_CONCAT(oralEvidence," ") AS oralEvidence FROM witnssUttrnces WHERE person IN ("Emma Carr", "Timo Hannay") GROUP BY person ORDER BY person')

##clear data and produce tdm for Emma Carr and Timo Hannay

##pa#ckages required: tm(corpus and cleaning) and textstem(lemmatize\_strings)

#THAndECUttrnces\_cor\_cl <- Corpus(VectorSource(THAndECUttrnces$oralEvidence))

#THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, tolower)

#stop.word <- unlist(read.table("stop\_word.txt", stringsAsFactors=FALSE))

#THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, removeWords, stop#.word)

#THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, removeNumbers)

#THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, removeWords, stopwords())

#THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, removePunctuation)

#stop.char <- unlist(read.table("stop\_char.txt", stringsAsFactors=FALSE))

#THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, removeWords, stop.char)

#THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, stripWhitespace)

#THAndECUttrnces\_cor\_cl <- tm\_map(THAndECUttrnces\_cor\_cl, lemmatize\_strings)

#THAndECUttrnces\_tdm\_cl <- TermDocumentMatrix(THAndECUttrnces\_cor\_cl)

colnames(THAndECUttrnces\_tdm\_cl) <- c("Emma Carr", "Timo Hannay")

#THAndECUttrnces\_tdm\_cl

## coerce as a matrix

#THAndECUttrnces\_tdm\_cl <- as.matrix(THAndECUttrnces\_tdm\_cl)

#colnames(THAndECUttrnces\_tdm\_cl) <- c("Emma Carr", "Timo Hannay")

## assign a palette

## package RColorBrewer is required

#if (!require('RColorBrewer')) install.packages('RColorBrewer'); library(RColorBrewer)

#pal <- brewer.pal(5, "Accent")

# plot wordclouds

#package wordcloud is required

#if (!require('wordcloud')) install.packages('wordcloud'); library(wordcloud)

#set.seed(12345)

#comparison.cloud(THAndECUttrnces\_tdm\_cl, scale=c(2,0.5), max.words = 50, rot.per = 0.3, random.order=FALSE, color=pal, title.size=2)

#set.seed(12345)

#commonality.cloud(THAndECUttrnces\_tdm\_cl, scale=c(5,0.5), max.words = 50, rot.per = 0.3, random.order=FALSE, color=pal, title.size=2)

#----------------------------------------

#code 07:

#use functions to reduce repetition of code

#7.1 function for forming a corpus and then cleaning and lemmatising it

#packages required: tm (corpus and cleaning) and textstem (lemmatize\_strings)

getFrmSctorUttrnces\_cleanNLemmatsdCrpus <- function(sctorUttrnces){

sctorUttrnces\_corpus <- Corpus(VectorSource(sctorUttrnces$oralEvidence))

sctorUttrnces\_corpus <- tm\_map(sctorUttrnces\_corpus, tolower)

stop.word <- unlist(read.table("stop\_word.txt", stringsAsFactors=FALSE))

sctorUttrnces\_corpus <- tm\_map(sctorUttrnces\_corpus, removeWords, stop.word)

sctorUttrnces\_corpus <- tm\_map(sctorUttrnces\_corpus, removeNumbers)

sctorUttrnces\_corpus <- tm\_map(sctorUttrnces\_corpus, removeWords, stopwords())

sctorUttrnces\_corpus <- tm\_map(sctorUttrnces\_corpus, removePunctuation)

stop.char <- unlist(read.table("stop\_char.txt", stringsAsFactors=FALSE))

sctorUttrnces\_corpus <- tm\_map(sctorUttrnces\_corpus, removeWords, stop.char)

sctorUttrnces\_corpus <- tm\_map(sctorUttrnces\_corpus, lemmatize\_strings)

sctorUttrnces\_corpus <- tm\_map(sctorUttrnces\_corpus, stripWhitespace)

return(sctorUttrnces\_corpus)

}

#7.2 function for producing a word cloud with a corpus

#package wordcloud is quired

prduceFrmSctorUttrnces\_corpus\_wordCloud <- function(sctorUttrnces\_corpus){

set.seed(12345)

wordcloud(sctorUttrnces\_corpus, scale=c(3,0.5), max.words = 100, rot.per = 0.3, random.order=FALSE, colors="black")

}

#7.3 function for producing a tdm with a corpus

prduceFrmSctorUttrnces\_corpus\_tdm <- function(sctorUttrnces\_corpus) return(TermDocumentMatrix(sctorUttrnces\_corpus))

#7.4 function for producing a csv file of word count list from a tdm

#the list will be sorted first according to scores (descending) and

#then according to alphabetical order (ascending)

prduceFrmSctorUttrnces\_tdm\_WordCntCsvFile <- function(sctorUttrnces\_tdm, fileNameIncldCsvSuffix){

sctorUttrnces\_wordCount <- as.data.frame(rowSums(as.matrix(sctorUttrnces\_tdm)))

sctorUttrnces\_wordCount <- cbind(rownames(sctorUttrnces\_wordCount), data.frame(sctorUttrnces\_wordCount, row.names = NULL))

colnames(sctorUttrnces\_wordCount) <- c('word', 'count')

sctorUttrnces\_wordCount <- sctorUttrnces\_wordCount[order(-sctorUttrnces\_wordCount[,2], sctorUttrnces\_wordCount[,1]),]

sctorUttrnces\_wordCount$rank <- seq.int(nrow(sctorUttrnces\_wordCount))

sctorUttrnces\_wordCount <- sctorUttrnces\_wordCount[, c('rank', 'word', 'count')]

write.table(sctorUttrnces\_wordCount, fileNameIncldCsvSuffix, quote = FALSE, sep = ',', row.name = FALSE)

}

#7.5 function for finding word frequencies directly from a corpus by

#using the freq\_terms function of the qdap package

#the top topHowMany words and words mast at least have atLeast characters

getFrmCorpus\_frequenciesOfWords <- function(corpus, topHowMany, atLeast) return(freq\_terms(as.data.frame(corpus), top = topHowMany, at.least = atLeast))

#7.6 function for producing an association word list from a

#dtm or a tdm

prduceFromDtmTdmAssocsWordsList <- function(dtmTdm, charVectWords, corLimitVect) return(findAssocs(dtmTdm, charVectWords, corLimitVect))

#7.7 function for producing a df from an association word list

produceFromAssocsWordsList\_df <- function(tdmDtm\_AssocsWordsList){

return(list\_vect2df(tdmDtm\_AssocsWordsList, col1 = 'word', col2 = 'assocsWord', col3 = 'corrScore'))

}

#----------------------------------------

#code 08:

#divide witnesses into the following four sectors:

##academic(Professor Yates, Dr McPherson, Professor Preston, Sir Nigel

#Shadbolt, Professor McAuley, Professor van Zoonen, Dr d'Aquin,

#Professor Robertson, Dr Macnish, Dr Elliot) total 10

##business(Sureyya Cansoy, Timo Hannay) total 2

##government(Steve Wood, Mr Vaizey) total 2

##ngo(Carl Miller, Emma Carr, Professor De Roure) total 3

#group utterances by sectors and treat replies to a question from a

#sector a document

#create a dataframe for each sector's utterances

academicUttrnces <- dbGetQuery(conndb1, "SELECT question, GROUP\_CONCAT(oralEvidence,' ') AS oralEvidence FROM witnssUttrnces JOIN persons ON witnssUttrnces.person = persons.person WHERE persons.sector = 'academic' GROUP BY question ORDER BY question")

businessUttrnces <- dbGetQuery(conndb1, "SELECT question, GROUP\_CONCAT(oralEvidence,' ') AS oralEvidence FROM witnssUttrnces JOIN persons ON witnssUttrnces.person = persons.person WHERE persons.sector = 'business' GROUP BY question ORDER BY question")

govtUttrnces <- dbGetQuery(conndb1, "SELECT question, GROUP\_CONCAT(oralEvidence,' ') AS oralEvidence FROM witnssUttrnces JOIN persons ON witnssUttrnces.person = persons.person WHERE persons.sector = 'government' GROUP BY question ORDER BY question")

ngoUttrnces <- dbGetQuery(conndb1, "SELECT question, GROUP\_CONCAT(oralEvidence,' ') AS oralEvidence FROM witnssUttrnces JOIN persons ON witnssUttrnces.person = persons.person WHERE persons.sector = 'ngo' GROUP BY question ORDER BY question")

#----------------------------------------

#code 09:

#performing text mining on dataframes created with code 08 above

#with functions shown on code 07 above

#9.1 code for producing a cleaned and lemmatised corpus with the

#utterances of witnesseses of each and every sector

#academic

academicUttrnces\_corpus <- getFrmSctorUttrnces\_cleanNLemmatsdCrpus(academicUttrnces)

#business

businessUttrnces\_corpus <- getFrmSctorUttrnces\_cleanNLemmatsdCrpus(businessUttrnces) #7.1

#government

govtUttrnces\_corpus <- getFrmSctorUttrnces\_cleanNLemmatsdCrpus(govtUttrnces)

#ngo

ngoUttrnces\_corpus <- getFrmSctorUttrnces\_cleanNLemmatsdCrpus(ngoUttrnces)

#9.2 code for producing a word cloud for every sector

#academic

prduceFrmSctorUttrnces\_corpus\_wordCloud(academicUttrnces\_corpus) #7.2

#business

prduceFrmSctorUttrnces\_corpus\_wordCloud(businessUttrnces\_corpus)

#government

prduceFrmSctorUttrnces\_corpus\_wordCloud(govtUttrnces\_corpus)

#ngo

prduceFrmSctorUttrnces\_corpus\_wordCloud(ngoUttrnces\_corpus)

#9.3 code for producing a tdm for every sector

#academic

academicUttrnces\_tdm <- prduceFrmSctorUttrnces\_corpus\_tdm(academicUttrnces\_corpus) #7.3

#business

businessUttrnces\_tdm <- prduceFrmSctorUttrnces\_corpus\_tdm(businessUttrnces\_corpus)

#government

govtUttrnces\_tdm <- prduceFrmSctorUttrnces\_corpus\_tdm(govtUttrnces\_corpus)

#ngo

ngoUttrnces\_tdm <- prduceFrmSctorUttrnces\_corpus\_tdm(ngoUttrnces\_corpus)

#9.4 code for producing a csv file of word freqency list for every sector

#academic

prduceFrmSctorUttrnces\_tdm\_WordCntCsvFile(academicUttrnces\_tdm, 'academicUttrncesWordCount.csv') #7.4

#business

prduceFrmSctorUttrnces\_tdm\_WordCntCsvFile(businessUttrnces\_tdm, 'businessUttrncesWordCount.csv')

#government

prduceFrmSctorUttrnces\_tdm\_WordCntCsvFile(govtUttrnces\_tdm, 'govtUttrncesWordCount.csv')

#ngo

prduceFrmSctorUttrnces\_tdm\_WordCntCsvFile(ngoUttrnces\_tdm, 'ngoUttrncesWordCount.csv')

#9.5 code for producing a word frequency list from each and every

#corpus instead of from tdm or dtm (results should be the same

#as those contained in the abovementioned csv files)

#academic

academic\_wordFrequency <- getFrmCorpus\_frequenciesOfWords(academicUttrnces\_corpus, 20, 3) #7.5

#business

business\_wordFrequency <- getFrmCorpus\_frequenciesOfWords(businessUttrnces\_corpus, 20, 3)

#government

govt\_wordFrequency <- getFrmCorpus\_frequenciesOfWords(govtUttrnces\_corpus, 20, 3)

#ngo

ngo\_wordFrequency <- getFrmCorpus\_frequenciesOfWords(ngoUttrnces\_corpus, 20, 3)

#forming a dataframe of the top 20 most frequent words of all sectors

#9.6 code for producing a dataframe which showing the

#top 20 most frequent words of each and every sector

word\_frequency\_top20\_df <- as.data.frame(cbind(academic\_wordFrequency[1:20,1], business\_wordFrequency[1:20,1], govt\_wordFrequency[1:20,1], ngo\_wordFrequency[1:20,1]))

names(word\_frequency\_top20\_df) <- c('academicSector', 'businessSector', 'govtSector', 'ngoSector')

word\_frequency\_top20\_df #show the df

#9.7 code for finding words in each and every corpus which are

#most correlated with datum, government, legislation and research

charVectWords <- c('datum', 'government', 'legislation', 'research')

corLimitVect <- c(0.4, 0.5, 0.5, 0.6)

#academic

academicUttrnces\_tdm\_AssocsWordsList <- prduceFromDtmTdmAssocsWordsList(academicUttrnces\_tdm, charVectWords, corLimitVect)

#business

businessUttrnces\_tdm\_AssocsWordsList <- prduceFromDtmTdmAssocsWordsList(businessUttrnces\_tdm, charVectWords, corLimitVect) #7.6

#government

govtUttrnces\_tdm\_AssocsWordsList <- prduceFromDtmTdmAssocsWordsList(govtUttrnces\_tdm, charVectWords, corLimitVect)

#ngo

ngoUttrnces\_tdm\_AssocsWordsList <- prduceFromDtmTdmAssocsWordsList(ngoUttrnces\_tdm, charVectWords, corLimitVect)

#9.8 code for producing a dataframe from the association word lists in respect the government sector corpus

govtUttrnces\_AssocsWords\_df <- produceFromAssocsWordsList\_df(govtUttrnces\_tdm\_AssocsWordsList) #7.7

#show part of the df: words associated with 'government'. corLimit = 0.5

govtUttrnces\_AssocsWords\_df[58:105,]

#----------------------------------------

#code 10:

#code for plotting a graph of those words associated with the word

#'government' with the correlation value equal or higher than 0.5 in the

#government sector utterances corpus

#packages ggplot2 and ggthemes are required

if (!require('ggplot2')) install.packages('ggplot2'); library(ggplot2)

if (!require('ggthemes')) install.packages('ggthemes'); library(ggthemes)

#use the list vector obtained from code 9.7 above

govtUttrnces\_tdm\_AssocsWords\_df\_government <- list\_vect2df(govtUttrnces\_tdm\_AssocsWordsList[2, drop = FALSE], col1 = "wordOfWhchAssWrdsTBFound", col2 = "assocsWords", col3 = "corrScore")

govtUttrnces\_tdm\_AssocsWords\_df\_government

#use the df obtained from the code above to plot the graph

govtUttrnces\_tdm\_AssocsWords\_df\_government <- govtUttrnces\_AssocsWords\_df[is.element(govtUttrnces\_AssocsWords\_df$word, 'government'),][,2:3]

ggplot(govtUttrnces\_tdm\_AssocsWords\_df\_government, aes(corrScore, assocsWord)) +

geom\_point(size = 3) + ggtitle('govtWtnsses\_OralEvidence\_WordsAssocsWth\_government') +

theme\_gdocs()

#----------------------------------------

#code 11

#attempt to use tf-idf to find which query or queires of

#the terms of reference each question is belonged to

#(see termsOfReference.csv for the six terms of reference)

#input, clean and lemmatise the terms of reference

termsOfReference\_df = read.csv('termsOfReference.csv', sep = '|', comment.char = '#', stringsAsFactors = FALSE)

termsOfReference\_list = as.list(termsOfReference\_df[,2])

termsOfReferenceCorpus = VectorSource(termsOfReference\_list)

termsOfReferenceCorpus\_preproc = Corpus(termsOfReferenceCorpus)

termsOfReferenceCorpus\_preproc = tm\_map(termsOfReferenceCorpus\_preproc,stripWhitespace)

termsOfReferenceCorpus\_preproc = tm\_map(termsOfReferenceCorpus\_preproc,removePunctuation)

termsOfReferenceCorpus\_preproc = tm\_map(termsOfReferenceCorpus\_preproc,content\_transformer(tolower))

termsOfReferenceCorpus\_preproc = tm\_map(termsOfReferenceCorpus\_preproc,removeWords,stopwords())

termsOfReferenceCorpus\_preproc = tm\_map(termsOfReferenceCorpus\_preproc, lemmatize\_strings)

inspect(termsOfReferenceCorpus\_preproc)

hc245\_tors = as.data.frame(termsOfReferenceCorpus\_preproc)

#optional: connect and inspect the database db1

#if table oralAll is not there, create one with code 02

#and code 03

#conndb1 <- dbConnect(RSQLite::SQLite(), 'db1')

#dbWriteTable(conndb1, 'oralAll', oralAll)

#create df of questions. Content of each question and its replies

#form a document

evidQsConcated <- dbGetQuery(conndb1, 'SELECT question, GROUP\_CONCAT(oralEvidence, " ") AS oralEvidence FROM oralAll GROUP BY question')

#form a named list of the questions

evidQsConcated\_list = as.list(evidQsConcated[,2])

evidQsConcated\_N.docs = length(evidQsConcated\_list)

names(evidQsConcated\_list) = evidQsConcated[,1]

#form a named list of the tors (cleaned and lemmatised version)

hc245\_tors\_list = unlist(hc245\_tors[,2])

hc245\_tors\_N.query = length(hc245\_tors\_list)

names(hc245\_tors\_list) = paste0("query", c(1:hc245\_tors\_N.query))

#move all to a corpus

evidQsConcated\_corpus = VectorSource(c(evidQsConcated\_list, hc245\_tors\_list))

evidQsConcated\_corpus$Names = c(names(evidQsConcated\_list),names(hc245\_tors\_list))

evidQsConcated\_corpus\_preproc = Corpus(evidQsConcated\_corpus)

#clean, trim and lemmatise the coprus

evidQsConcated\_corpus\_preproc = tm\_map(evidQsConcated\_corpus\_preproc,stripWhitespace)

evidQsConcated\_corpus\_preproc = tm\_map(evidQsConcated\_corpus\_preproc,removePunctuation)

evidQsConcated\_corpus\_preproc = tm\_map(evidQsConcated\_corpus\_preproc,content\_transformer(tolower))

evidQsConcated\_corpus\_preproc = tm\_map(evidQsConcated\_corpus\_preproc,removeWords,stopwords())

evidQsConcated\_corpus\_preproc = tm\_map(evidQsConcated\_corpus\_preproc, lemmatize\_strings)

#form term document matrix

evidQsConcated\_tdm = TermDocumentMatrix(evidQsConcated\_corpus\_preproc,control = list(weighting = function(x) weightTfIdf(x, normalize = FALSE)))

evidQsConcated\_tdm\_mat = as.matrix(evidQsConcated\_tdm)

colnames(evidQsConcated\_tdm\_mat) = c(names(evidQsConcated\_list),names(hc245\_tors\_list))

#normalising the tdm

evidQsConcated\_tfidf\_mat <- scale(evidQsConcated\_tdm\_mat, center = FALSE,scale = sqrt(colSums(evidQsConcated\_tdm\_mat^2)))

#split qsAndEvidence and tors

hc245\_tors.vectors <- evidQsConcated\_tfidf\_mat[, (evidQsConcated\_N.docs + 1):(evidQsConcated\_N.docs+hc245\_tors\_N.query)]

evidQsConcated\_tfidf\_mat <- evidQsConcated\_tfidf\_mat[, 1:evidQsConcated\_N.docs]

#calculate the similarity scores

evidQsConcated\_doc.scores <- t(hc245\_tors.vectors) %\*% evidQsConcated\_tfidf\_mat

#change the tors wording to original wording

hc245\_tors\_list = unlist(termsOfReference\_df[,2])

names(hc245\_tors\_list) = paste0("query", c(1:hc245\_tors\_N.query))

evidQsConcated\_results.df <- data.frame(querylist = hc245\_tors\_list,evidQsConcated\_doc.scores)

evidQsConcated\_showTopresults <- function(query, noOfDocs){

x = evidQsConcated\_results.df[which(evidQsConcated\_results.df$querylist == query),]

yy = data.frame(t(x),rownames(t(x)),row.names = NULL)[-1,]

names(yy) = c("score","docs")

yy$score = as.numeric(as.character(yy$score))

yyy = yy[order(yy$score,decreasing = T),]

return(yyy[which(yyy$score > 0),][1:noOfDocs,])

}

#find the top 20 questions which are,

#according to tf-idf, most related to terms of reference 1

evidQsConcated\_showTopresults("How can real-time analysis of social media data benefit the UK? What should the Government be doing to maximise these benefits?", 20)

#according to tf-idf, the top 20 questions which are

#most related to terms of reference 4

evidQsConcated\_showTopresults("What are the ethical concerns of using personal data and how is this data anonymised for research?", 20)

#please see termsOfReference.csv for the six terms of reference

#----------------------------------------

#code 12

#finding the p-value by simulation

zero1200One800 = append(rep(0, 1200), rep(1, 800))

countAtLeast <- function(smplPop, pickup, rept, threshold){

cnt2 = 0

for(i in 1: 100) {

cnt <- 0

for(i in 1: rept){

if(sum(sample(smplPop, pickup, replace = FALSE)) <= threshold) cnt <- cnt + 1

}

cnt2 = cnt2 + cnt

}

return(cnt2/100)

}

zero1200One800 = append(rep(0, 1200), rep(1, 800))

p.value <- countAtLeast(zero1200One800, 17, 100, 4)

p.value

#----------------------------------------

#code 13

#Chi-squared test (independence test, by simulation and by the fisher method)

atttdeToEULegsntnInput =("

Sector,Positive,Neutral,Negative

academic,1,6,3

business,0,0,2

government,2,0,0

ngo,1,1,1

")

atttdeToEULegsntnMtrx = as.matrix(read.table(textConnection(atttdeToEULegsntnInput), sep = ',', header=TRUE, row.names=1))

atttdeToEULegsntnMtrx

atttdeToEULegsntnChisq <- chisq.test(atttdeToEULegsntnMtrx)

atttdeToEULegsntnChisq

atttdeToEULegsntnChisqSimu <- chisq.test(atttdeToEULegsntnMtrx, simulate.p.value = TRUE)

atttdeToEULegsntnChisqSimu

atttdeToEULegsntnFisher <- fisher.test(atttdeToEULegsntnMtrx)

atttdeToEULegsntnFisher

#----------------------------------------

#code 14

#Chi-squared test (one homogeneity test and two independence tests)

theme\_evidencePara <- read.csv('theme\_evidencePara.csv', stringsAsFactors = FALSE)

theme\_reportPara <- read.csv('theme\_reportPara.csv', stringsAsFactors = FALSE)

responsibleUseOfData <- read.csv('u1720146\_CN8001\_Responsible\_Use\_of\_Data.csv', comment.char = '#', quote = '^"\'', sep = '|', stringsAsFactors = FALSE)

dbWriteTable(conndb1, 'theme\_evidencePara', theme\_evidencePara, overwrite = TRUE)

dbWriteTable(conndb1, 'theme\_reportPara', theme\_reportPara, overwrite = TRUE)

dbWriteTable(conndb1, 'responsibleUseOfData', responsibleUseOfData, overwrite = TRUE)

dbListTables(conndb1) #inspect db1

themeEvidenceNoOfOwrds <- dbGetQuery(conndb1, "SELECT theme, ROUND(SUM(wholeOrHalf \* (LENGTH(TRIM(oralEvidence)) - LENGTH(TRIM(REPLACE(oralEvidence, ' ', ''))) + 1))) AS 'totalNoOfWords(Evidence)' FROM oralAll JOIN theme\_evidencePara ON id = paraNo GROUP BY theme ORDER BY CASE WHEN theme = 'benefits' THEN 1 ELSE theme END") #41914 correct!

themeReportNoOfWords <- dbGetQuery(conndb1, "SELECT theme, SUM(LENGTH(TRIM(content)) - LENGTH(TRIM(REPLACE(content, ' ', ''))) + 1) AS 'totalNoOfWords(Report)' FROM theme\_reportPara JOIN responsibleUseOfData ON theme\_reportPara.paraNo = responsibleUseOfData.paraNo GROUP BY theme ORDER BY CASE WHEN theme = 'benefits' THEN 1 ELSE theme END") #6604 correct!

#produce an observed count table of distribution of words

themeEvidenceReportWordsDf <- cbind(themeEvidenceNoOfOwrds, themeReportNoOfWords)

rownames(themeEvidenceReportWordsDf) <- themeEvidenceReportWordsDf[,1]

themeEvidenceReportWordsDf <- themeEvidenceReportWordsDf[,c(2,4)]

evidenceReportThemeWordsDf <- t(themeEvidenceReportWordsDf)

evidenceReportThemeWordsDf

#chi-squared test (homogeneity test)

#null hypothesis: difference in distributions of words across the themes are not significant between the witnesses' evidence and the report

chisq.test(evidenceReportThemeWordsDf)

sectorThemeNoOfWords <- dbGetQuery(conndb1, "WITH oralAllWithSector AS (SELECT id, sector, LENGTH(TRIM(oralEvidence)) - LENGTH(TRIM(REPLACE(oralEvidence, ' ', ''))) + 1 AS noOfWords FROM oralAll JOIN persons WHERE oralAll.person = persons.person) SELECT sector, theme, ROUND(sum(noOfWords \* wholeOrHalf)) AS noOfWords FROM oralAllWithSector JOIN theme\_evidencePara WHERE id = paraNo GROUP BY sector, theme") #41913 correct!

sectorThemeNoOfWords

sectorThemeNoOfWordsTable <- xtabs(noOfWords~sector+theme,data = sectorThemeNoOfWords)

sectorThemeNoOfWordsTable

#chi-squared test (independence test)

#null hypothesis: career sector persons belong to and the length of their utterances allocated to the four themes are independent

chisq.test(sectorThemeNoOfWordsTable)

panelThemeNoOfWords <- dbGetQuery(conndb1, "WITH oralAllWithPanel AS (SELECT id, panel, (LENGTH(TRIM(oralEvidence)) - LENGTH(TRIM(REPLACE(oralEvidence, ' ', ''))) + 1) AS noOfWords FROM oralAll JOIN persons WHERE oralAll.person = persons.person) SELECT panel, theme, ROUND(SUM(noOfWords \* wholeOrHalf)) AS noOfWords FROM oralAllWithpanel JOIN theme\_evidencePara WHERE id = paraNo GROUP BY panel, theme")

panelThemeNoOfWords

panelThemeNoOfWordsTable <- xtabs(noOfWords~panel+theme,data = panelThemeNoOfWords)

panelThemeNoOfWordsTable

#chi-squared test (independence test)

#null hypothesis: lengths of utterances allocated to the four themes were independent to the panels

chisq.test(panelThemeNoOfWordsTable)