WSP R Open Question Analysis

Lizzie Jones[1]

02/05/2021

Text analysis and visualisations for open questions

This rMarkdown explores and analyses the open questions using text, work frequency and sentiment analysis techniques which are beyond the scope of NVivo (or where R is more effective).

The key open-ended questions include:

- Q8. How did you feel when you saw WS in the wild?
- Q11a-c. Three words used to describe white storks
- Q15. Do you support the White Stork Project?
- Q16. What are yours views on the management of White Storks?

I clean and explore each question in turn in the sections below and comment on any interesting findings.

Q11. What descriptive words do you associate with white storks?

```
# Q11 (words to describe WSP)
#Clean text
head(words_df)
```

```
## # A tibble: 6 x 4
##   UniqueID_short SurveyType Word_num  Words
##            <int> <fct>      <chr>     <fct>
## 1              1 Proactive  Q11_word1 "One leg"
## 2              1 Proactive  Q11_word2 "Babies"
## 3              1 Proactive  Q11_word3 ""
## 4              2 Proactive  Q11_word1 "White"
## 5              2 Proactive  Q11_word2 "Long legs"
## 6              2 Proactive  Q11_word3 "A bird"
```

```
words_df$Words <- gsub("[^[:graph:]]", " ", words_df$Words) #get rid of non graphical characters
words_df$Words <- gsub("rt", "", words_df$Words)# Replace blank space ("rt")
words_df$Words <- gsub("[[:punct:]]", "", words_df$Words)# Remove punctuation
words_df$Words <- gsub("[ |\t]{2,}", "", words_df$Words)# Remove tabs
words_df$Words <- gsub("^ ", "", words_df$Words)# Remove blank spaces at the beginning
words_df$Words <- gsub(" $", "", words_df$Words)# Remove blank spaces at the end
words_df$Words <- tolower(words_df$Words)#convert all text to lower case

Corpus_words <- Corpus(VectorSource(words_df$Words))
Corpus_words <- tm_map(Corpus_words, removeNumbers)
Corpus_words <- tm_map(Corpus_words, removeWords, stopwords("english")) #removes common english stopwords
# Corpus_words <- tm_map(Corpus_words, removeWords, c("muffin"))  #You can specify words to remove
# Corpus_words <- tm_map(Corpus_words, PlainTextDocument)

#build a term-document matrix
library("tm")
TDM_words = tm::TermDocumentMatrix(Corpus_words, control = list(minWordLength = 1))
m = as.matrix(TDM_words)
v = sort(rowSums(m), decreasing = TRUE)
d = data.frame(word = names(v),freq=v)

# Create a wordcloud
wordcloud(Corpus_words, scale=c(5,0.5), max.words=100, random.order=FALSE, rot.per=0.25,
          use.r.layout=FALSE, colors=brewer.pal(10,"Spectral"))
```

Word frequency analysis (Words used to describe White Storks)

```
# Frequent word analysis
# We can find the words that appear at least 100 times by calling the findFreqTerms() function on the term.doc.ma
trix
HiFreq_words <- findFreqTerms(TDM_words, 100)
HiFreq_words
```

```
##  [1] "babies"      "white"       "long"        "bird"        "rare"
##  [6] "large"       "big"         "majestic"    "graceful"    "beautiful"
## [11] "elegant"     "tall"        "impressive"  "interesting"
```

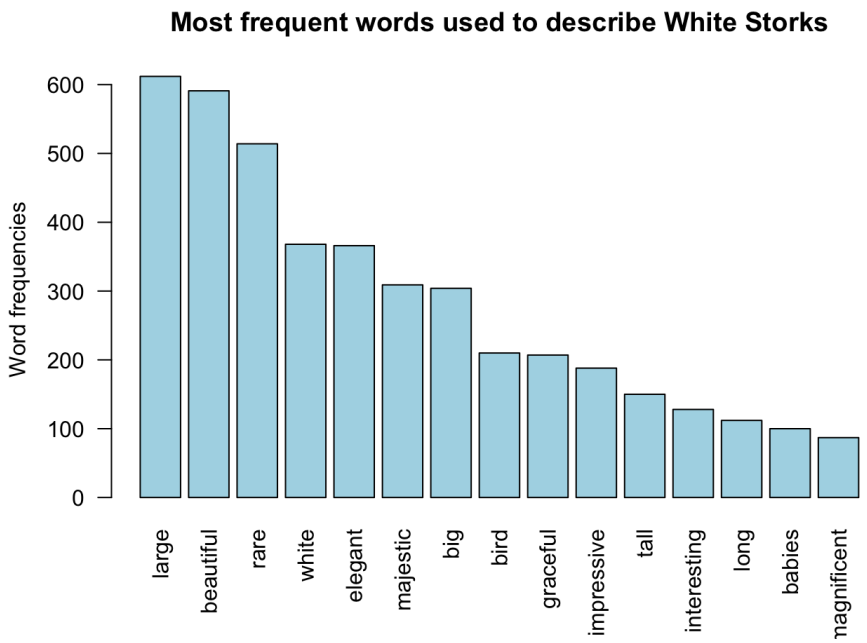
```
# Now you also see how associated a word is to another word or a list of words.
findAssocs(TDM_words, HiFreq_words, 0.4)
```

```
## $babies
## numeric(0)
##
## $white
## numeric(0)
##
## $long
## legs
## 0.51
##
## $bird
## numeric(0)
##
## $rare
## numeric(0)
##
## $large
## numeric(0)
##
## $big
## numeric(0)
##
## $majestic
## numeric(0)
##
## $graceful
## numeric(0)
##
## $beautiful
## numeric(0)
##
## $elegant
## numeric(0)
##
## $tall
## numeric(0)
##
## $impressive
## numeric(0)
##
## $interesting
## numeric(0)
```

```
# or, just compute word strength associations
findAssocs(TDM_words, "long", 0.5) # Looks like the word “long” and “legs” are very frequently associated (51% of the time)
```

```
## $long
## legs
## 0.51
```

```
barplot(d[1:15,]$freq, las = 2, names.arg = d[1:15,]$word,
        col ="lightblue", main ="Most frequent words used to describe White Storks",
        ylab = "Word frequencies")
```



Q8. How did you feel when you saw WS in the wild?

```
# Polarity / Sentiment Analysis

### Q8. How did you feel when you saw WS in the wild?
head(final_data$Q8.2_feelings)
```

```
## [1] <NA>
## [2] I saw them nesting on cliff tops and rocks surrounded by the sea in Portugal - it was really cool!
## [3] always a pleasing sight, no matter how many you've seen already
## [4] They are common throughout many parts of Europe so didn't feel anything in particular but would be ecstati
c to see one over London.
## [5] Fascinated, and in awe. They're size when flying over head was outstanding (made all the more incredible w
ith a huge feather dropping by my feet!) not something I imagined experiencing in the UK. My first experience of
them was in Hungary, when I heard their bizarre clacking and wondered what on earth it was. Soon I saw them nesti
ng on the chimneys and poles in towns and on roads, their clacking gave the soundtrack to the area a 'wild' sens
e.
## [6] <NA>
## 984 Levels:   Seen them in France. Excited! ... Wow. I wish I could see these in the UK.
```

```
# Clean the data
final_data$Q8.2_feelings_text <- gsub("[^[:graph:]]", " ", final_data$Q8.2_feelings) #get rid of non graphical ch
aracters
final_data$Q8.2_feelings_text <- gsub("^ ", "", final_data$Q8.2_feelings_text)# Remove blank spaces at the beginn
ing
# Sentiment
class(final_data$Q8.2_feelings_text)
```

```
## [1] "character"
```

```
sentiment(get_sentences(final_data$Q8.2_feelings_text))
```

```
##       element_id sentence_id word_count  sentiment
##    1:          1           1         NA  0.0000000
##    2:          2           1         19  0.5391270
##    3:          3           1         11  0.3015113
##    4:          4           1         23 -0.1907907
##    5:          5           1          4  0.8000000
##   ---
## 3897:       3536           1         NA  0.0000000
## 3898:       3537           1          1  0.7500000
## 3899:       3537           2          6  0.2041241
## 3900:       3538           1         NA  0.0000000
## 3901:       3539           1         NA  0.0000000
```

```
Corpus_feelings <- Corpus(VectorSource(final_data$Q8.2_feelings_text))
Corpus_feelings <- tm_map(Corpus_feelings, removeNumbers)
Corpus_feelings <- tm_map(Corpus_feelings, removeWords, stopwords("english")) #removes common english stopwords
Corpus_feelings <- tm_map(Corpus_feelings, removeWords, c("they", "the", "also", "I'm", "don't", "can"))  #You ca
n specify words to remove

#build a term-document matrix
TDM_feelings = tm::TermDocumentMatrix(Corpus_feelings, control = list(minWordLength = 1))
m_feelings = as.matrix(TDM_feelings)
v_feelings = sort(rowSums(m_feelings), decreasing = TRUE)
d_feelings = data.frame(word = names(v_feelings),freq=v_feelings)

# Create a wordcloud
wordcloud(Corpus_feelings, scale=c(5,0.5), max.words=100, random.order=FALSE, rot.per=0,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```

Q15a. Support for WS reintroduction

Question = Do you support the reintroduction of white Storks to southern England?

```
### Q15. Do you support the WSP?
head(final_data$Q15_WSP_support_open)
```

```
## [1] It's always good to have as much diverse life as possible, and if they used to strive here, why not again?
If handled correctly of course.
## [2]
## [3]
## [4] The more rewilding the better.
## [5] I absolutely support this, however it does concern me that they're reliant upon wetland ecosystems, which
we have so little of. It's a natural follow on to the reintroduction of the beaver of course, and imagining beave
r wetlands with white storks feeding within them is thrilling! But the widespread (government approved) support o
f free-living beavers seems to be at a much slower pace than the potential speed of breeding and dispersal of whi
te storks. But I'm all for bringing appropriate species back, like the white stork, asap despite this.
## [6]
## 1910 Levels:  ...
```

```
# Clean the data
final_data$Q15_WSP_support_text <- gsub("[^[:graph:]]", " ", final_data$Q15_WSP_support_open) # get rid of non gr
aphical characters
final_data$Q15_WSP_support_text <- gsub(",", " ", final_data$Q15_WSP_support_open) # Remove commas after words
final_data$Q15_WSP_support_text <- gsub("'", "", final_data$Q15_WSP_support_open) # Remove apostrophes
final_data$Q15_WSP_support_text <- gsub("^ ", "", final_data$Q15_WSP_support_text) # Remove blank spaces at the b
eginning
final_data$Q15_WSP_support_text <- gsub(" $", "", final_data$Q15_WSP_support_text) # Remove blank spaces at the e
nd

# Reasons for support/not support WSP
class(final_data$Q15_WSP_support_text)
```

```
## [1] "character"
```

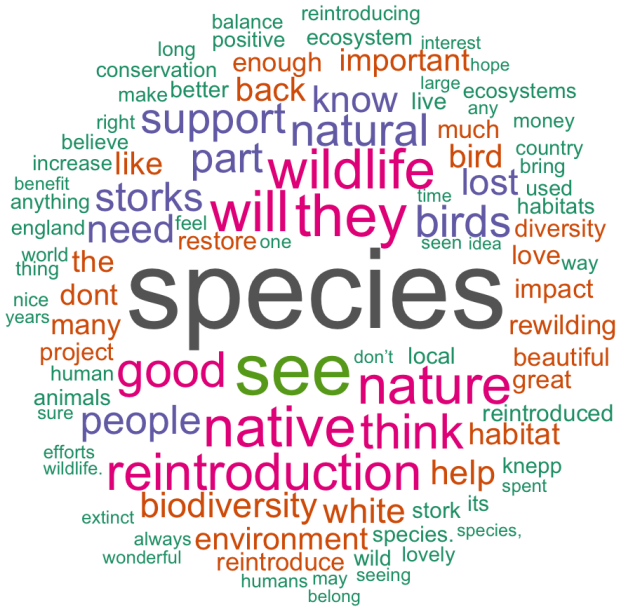
```
sentiment(get_sentences(final_data$Q15_WSP_support_text))
```

```
##       element_id sentence_id word_count   sentiment
##    1:          1           1         21  0.29459415
##    2:          1           2          5  0.35777088
##    3:          2           1         NA  0.00000000
##    4:          3           1         NA  0.00000000
##    5:          4           1          5  0.64398758
##   ---
## 4162:       3535           1         NA  0.00000000
## 4163:       3536           1         14 -0.32071349
## 4164:       3537           1          6  0.44907312
## 4165:       3538           1         14  0.05345225
## 4166:       3539           1          4  0.00000000
```

```
Corpus_support <- Corpus(VectorSource(final_data$Q15_WSP_support_text))
Corpus_support <- tm_map(Corpus_support, removeNumbers)
Corpus_support <- tm_map(Corpus_support, removeWords, stopwords("english")) #removes common english stopwords
Corpus_support <- tm_map(Corpus_support, removeWords, c("they", "the", "also", "I'm", "don't", "can"))  #You can
 specify words to remove

#build a term-document matrix
TDM_support = tm::TermDocumentMatrix(Corpus_support, control = list(minWordLength = 1))
m_support = as.matrix(TDM_support)
v_support = sort(rowSums(m_support), decreasing = TRUE)
d_support = data.frame(word = names(v_support),freq=v_support)

# Create a wordcloud
wordcloud(Corpus_support, scale=c(5,0.5), max.words=100, random.order=FALSE, rot.per=0,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```




```
# There are lots more ways of doing this (see the QDAP package vignette). Here we take a cleaned character vector
used earlier (i.e. words_df$Words) and compare its sentiment against a grouping variable (e.g. SurveyType)
# poldat_surveytype <- with(all_data, polarity(words_df$Words, all_data$SurveyType))
# plot(poldat)
```

Q16a. Expressing views on WS management

Question = Do you feel that you can express your views on the ongoing white stork reintroduction in a way that will influence management decisions?

```
# Polarity / Sentiment Analysis

### Q16. What are yours views on the management of White Storks?
head(final_data$Q16_views_management_open)
```

```
## [1]
## [2]
## [3] question seems unclear
## [4] I work in conservation and rewilding so am knowledgable and trained on the subject.
## [5] I've not looked into the project as much as I should, which is dreadful of me. To be honest I think that's
because I live in the north, and we forever seem to be far behind in the progress of bold reintroductions - even
free-living beavers are barely discussed here, so I suppose it seems a long way off to have White Storks (though
I hope not).
## [6]
## 1297 Levels:  ...
```

```
# Clean the data
final_data$Q16_views_management_text <- gsub("[^[:graph:]]", " ", final_data$Q16_views_management_open) #get rid
 of non graphical characters
final_data$Q16_views_management_text <- gsub("^ ", "", final_data$Q16_views_management_text)# Remove blank spaces
at the beginning
final_data$Q16_views_management_text <- gsub(" $", "", final_data$Q16_views_management_text)# Remove blank spaces
at the end

# Reasons for support/not support WSP
class(final_data$Q16_views_management_text)
```

```
## [1] "character"
```

```
sentiment(get_sentences(final_data$Q16_views_management_text))
```

```
##       element_id sentence_id word_count  sentiment
##    1:          1           1         NA  0.0000000
##    2:          2           1         NA  0.0000000
##    3:          3           1          3 -0.2886751
##    4:          4           1         14  0.2806243
##    5:          5           1         16 -0.1875000
##   ---
## 3804:       3535           1         NA  0.0000000
## 3805:       3536           1         13  0.0000000
## 3806:       3537           1         NA  0.0000000
## 3807:       3538           1          3  0.0000000
## 3808:       3539           1          4  0.0000000
```

```
# Wrd frequencies
Corpus_management <- Corpus(VectorSource(final_data$Q16_views_management_text))
Corpus_management <- tm_map(Corpus_management, removeNumbers)
Corpus_management <- tm_map(Corpus_management, removeWords, stopwords("english")) #removes common english stopwor
ds
Corpus_management <- tm_map(Corpus_management, removeWords, c("they", "the", "also"))  #You can specify words to
 remove

#build a term-document matrix
TDM_management = tm::TermDocumentMatrix(Corpus_management, control = list(minWordLength = 1))
m_management = as.matrix(TDM_management)
v_management = sort(rowSums(m_management), decreasing = TRUE)
d_management = data.frame(word = names(v_management),freq=v_management)

# Create a wordcloud
wordcloud(Corpus_management, scale=c(5,0.5), max.words=100, random.order=FALSE, rot.per=0,
          use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```

1. University of Brighton, l.jones4@brighton.ac.uk (mailto:l.jones4@brighton.ac.uk)↩