What is a Corpus?

Kevan Rajaram

5/3/2020

No no - not corpse! Corpus. It is a Latin word meaning body and was popularized in the 18th Century. We will being using it’s Oxford dictionary’s meaning: “a collection of written texts, especially the entire works of a particular author or a body of writing on a particular subject.”

Our corpus is the Old Testament data we scraped from <https://www.bible.com>.  
If you haven’t read that - take a look at it [here](https://www.aimtt.co/in-the-beginning)

Well let’s not waste any time. This blog’s objective is to analyze that Old Testament data and we will create those questions now:  
1. What is/are the most used word/words in the Old Testament?  
2. Are the words associated in any way?  
3. Can we can summarize the text data into a single/couple comprehensive sentences?

Most of my analysis comes from a very nifty R package called **“tm”** which allows **t**ext **m**ining in R. More on [tm](https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf) here.

require(tidyr)  
require(dplyr)  
require(xlsx)  
setwd("C:/Users/krajaram/Desktop/Personal/AIM BLOG/03. What-s A Corpus")  
textdata1 = read.csv("bibletextdata.csv", header=TRUE, encoding = "UFT-8")  
textdata1 <- as.matrix(textdata1$bibletextdata)

library(tm)  
  
biblecorpus <- VCorpus(VectorSource(textdata1))  
biblecorpusclean <- tm\_map(biblecorpus,content\_transformer(tolower)) #converting to lower case letters  
biblecorpusclean <- tm\_map(biblecorpusclean,removeNumbers) #removing numbers  
biblecorpusclean <- tm\_map(biblecorpusclean,removeWords,c(stopwords('english'))) #removing common english words  
biblecorpusclean <- tm\_map(biblecorpusclean,removePunctuation)

As usual we have some pre-processing of data to do. That chunk of code above is doing just that. Firstly I converted my data frame to a so-called Corpus held in a VCorpus (Volatile Corpus) - recognized as volatile since corpora (plural of corpus - it was new to me too) which are stored as R objects in memory and can be destroyed. There are other types that can be reviewed.

What does this clean corpus look like? What did the original corpus look like?

inspect(biblecorpus[1])

## <<VCorpus>>  
## Metadata: corpus specific: 0, document level (indexed): 0  
## Content: documents: 1  
##   
## [[1]]  
## <<PlainTextDocument>>  
## Metadata: 7  
## Content: chars: 4119

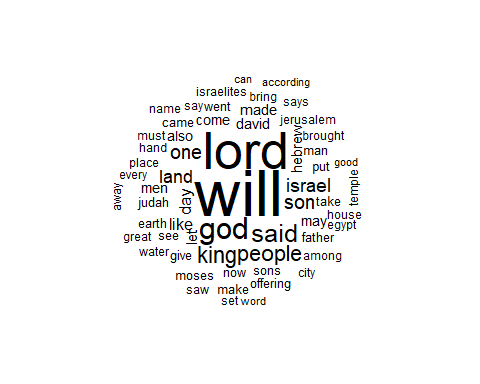
inspect(biblecorpusclean[1])

## <<VCorpus>>  
## Metadata: corpus specific: 0, document level (indexed): 0  
## Content: documents: 1  
##   
## [[1]]  
## <<PlainTextDocument>>  
## Metadata: 7  
## Content: chars: 2741

Using Chapter 1 of of Genesis in the Old Testament as an example, shown above, we can see the clean corpus (removing numbers, special characters, punctuation marks, etc) and converting it all to lower case text to perform our analyses has greatly reduced the amount of characters (approximately 50%). It should be in better shape to start analyzing now.

Well let us get a sense of the most frequent words.  
Below the word cloud lines will do just that. I set a minimum threshold of 500 as the frequency requirement to show up in this diagram. This should paint a good sense of what’s a popular word or not.

library(wordcloud)  
wordcloud(biblecorpusclean , min.freq = 500, random.order = FALSE)



Well **“Will”** pops very dramatically - perhaps **‘Will’** is a central theme of the Old Testament. Other noticeable words which are fairly obvious are **Lord**, **God** , **Said**, **Son**, **Israel**, etc..  
What else can we find?

How about word associativity? That is, which words are highly correlated with other words. We can use the most frequent terms above to see what other words they are most highly correlated with. This paints an even better picture of what is going on.  
Let’s look for a high degree of correlation - one that gives us confidence in the assumption - Let’s say about 70% correlation or 0.7.

bibletdm <- TermDocumentMatrix(biblecorpusclean, control = list(wordLengths = c(1, Inf)))  
#freq.terms <- findFreqTerms(bibletdm, lowfreq=500) #find frequency  
#assoc = findAssocs(bibletdm, freq.terms, 0.7) #create list of correlations at 0.7 for frequent terms

**Note: You can remove the comments in the above code to frequency as a list and the associations at correlation of 0.7**

What words are specifically associated with **will** and **God**?

print(findAssocs(bibletdm, "will", 0.7))

## $will  
## numeric(0)

print(findAssocs(bibletdm, "god", 0.7))

## $god  
## morning— sky ground—everything kinds   
## 0.78 0.78 0.77 0.77   
## seedbearing teems vault created   
## 0.77 0.77 0.77 0.76   
## hovering lights lights— teem   
## 0.76 0.76 0.76 0.76   
## vegetation winged bearing formless   
## 0.76 0.76 0.75 0.75   
## govern increase lesser separate   
## 0.75 0.75 0.75 0.75   
## light seed separated move   
## 0.74 0.74 0.74 0.73   
## plants subdue birds   
## 0.73 0.71 0.70

At a 0.7 level of correlation ‘will’ seems to have no associations but ‘God’ has quite a few - such as “day”, “land”, night" , “seas” , “sky”. God seems to have a high association with the elements - I mean He did create everything after all.

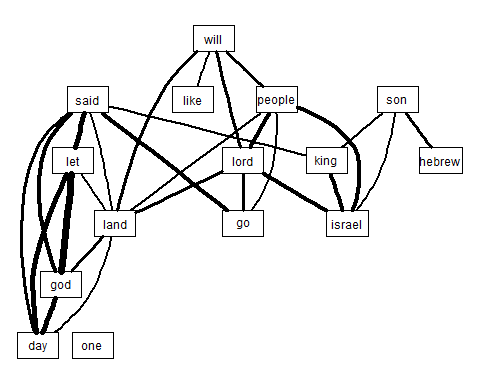
Perhaps because ‘will’ is so frequent it is difficult to associate words at a high correlation - so perhaps we should lower the correlation to a lower lever than what it currently is and see if it produces any results.

print(findAssocs(bibletdm, "will", 0.35))

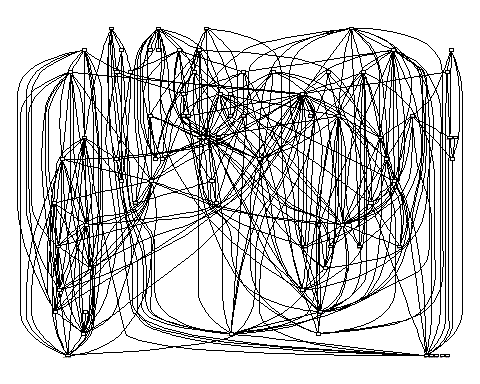
## $will  
## fall nations says afflict declares therefore   
## 0.40 0.40 0.40 0.37 0.37 0.36

Yes! Results. We can take a look at the word associativity for ‘will’ but at a much lower level. Take a look and see if you can gather any interesting findings.  
How about a generalized view of word associations? Does it give a complex pattern?  
There are a lot of ways one can look at this for example we can increase the frequency threshold to have very frequent words but lower the correlation threshold to see associativity with weighting applicable **OR** there can be a decreased frequency threshold with the same correlation to see associativity with weighting still applicable **OR** other combinations of this methodology.

freq.terms1 <- findFreqTerms(bibletdm, lowfreq=1000)  
plot(bibletdm, term = freq.terms1 , corThreshold = 0.2, weighting = T, cex=1)



freq.terms2 <- findFreqTerms(bibletdm, lowfreq=400)  
plot(bibletdm, term = freq.terms2 , corThreshold = 0.2, weighting = F, cex=1)



Examine the output above with the density of the line in the first chart implying stronger correlation/associativity.  
The first chart had a higher barrier to entry so the network there is pretty clear with the latter being the opposite with a more complex network (hence it looks so jumbled).

Again from the first graph above, we see **will** strongly related to **lord**, **people** and **land**. Very interesting thoughts and ideas.

Another interesting thing may be for us to study a string of words or n-gram words that have been tokenized. We already have this data in the form of *‘biblecorpusclean’*. Let’s take an example of **God** to see how how it performs with tokens of 2 and 3 to see what we can deduce about God as a series of words.

library(corpus)  
term\_stats(biblecorpusclean, ngrams = 2:3, types = TRUE,  
 subset = type1 == "god" & !type2 %in% stopwords\_en)

## term type1 type2 type3 count support  
## 1 god israel god israel <NA> 198 125  
## 2 god said god said <NA> 291 61  
## 3 god made god made <NA> 114 53  
## 4 god lord god lord <NA> 53 45  
## 5 god israel says god israel says 62 43  
## 6 god god god god <NA> 41 38  
## 7 god blessed god blessed <NA> 64 37  
## 8 god called god called <NA> 87 33  
## 9 god ancestors god ancestors <NA> 41 32  
## 10 god set god set <NA> 32 32  
## 11 god saw god saw <NA> 192 29  
## 12 god created god created <NA> 111 29  
## 13 god said let god said let 217 28  
## 14 god brought god brought <NA> 30 28  
## 15 god created male god created male 28 28  
## 16 god created mankind god created mankind 28 28  
## 17 god made two god made two 28 28  
## 18 god saw good god saw good 135 27  
## 19 god blessed said god blessed said 54 27  
## 20 god called dry god called dry 27 27  
## <U+22EE>(2937 rows total)

As you can see from the above output there are lots of 2-gram and 3-gram sequences related to “God” - one of the most popular ones based on count and support are “God said” and “God said let”. Shows how that God spoke a great deal in the Old Testament and He also instructed a lot. Very great points!

The last thing I’d like for us to challenge is a key-takeaway or summary of the bible text before we converted it to a clean corpus because here the tokens (words) and punctuation marks matter for us to construct indices so we can deduce as an encompassing summary.  
This is done using a vector of documents using the page rank algorithm or degree centrality which is further explained [here](http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume22/erkan04a-html/erkan04a.html).

require(lexRankr)  
test = textdata1[1:50,] ##Testing with Genesis only  
  
#perform lexrank for top 3 sentences  
top\_3 = lexRankr::lexRank(test,  
 #only 1 article; repeat same docid for all of input vector  
 docId = rep(1, length(test)),  
 #return 3 sentences to mimick autotldr's output  
 n = 3,  
 continuous = TRUE)

## Parsing text into sentences and tokens...DONE  
## Calculating pairwise sentence similarities...DONE  
## Applying LexRank...DONE  
## Formatting Output...DONE

#reorder the top 3 sentences to be in order of appearance in article  
order\_of\_appearance = order(as.integer(gsub("\_","",top\_3$sentenceId)))  
ordered\_top\_3 = top\_3[order\_of\_appearance, "sentence"] #extract sentences in order of appearance  
ordered\_top\_3

## [1] "7The Lord, the God of heaven, who brought me out of my father’s household and my native land and who spoke to me and promised me on oath, saying, To your offspring#24:7 Or seed I will give this land’—he will send his angel before you so that you can get a wife for my son from there."   
## [2] "When Esau left for the open country to hunt game and bring it back, 6Rebekah said to her son Jacob, Look, I overheard your father say to your brother Esau, 7Bring me some game and prepare me some tasty food to eat, so that I may give you my blessing in the presence of the Lord before I die.’ 8Now, my son, listen carefully and do what I tell you: 9Go out to the flock and bring me two choice young goats, so I can prepare some tasty food for your father, just the way he likes it."  
## [3] "9Now hurry back to my father and say to him, This is what your son Joseph says: God has made me lord of all Egypt."

Well that was again another interesting piece of code but follow it and you would see the above was a test on the text summary from Genesis. From the method, we see the lexical rank of these sentences are all key in the Old Testament and summarizes keys aspects of what formed key covenants between God and His people including Abraham/Isaac, Jacob and Joseph. These are what our model constructed as lexically centralized. It may vary to what you or I think but remember the model is performing a graph based method of relative importance.

There are so many other interesting types of analyses we can now construct but this was just an introduction into this world of text mining and NLP. I hope you enjoyed it and you take the code and try manipulating it for yourself to see variations of outcomes.

***For my next entry, I’d be looking at predictive modelling! Stay tuned!***