**Introduction to Corpus Linguistics**

SoSe 2018-2019

by Dr. Natalia Levshina ©

**Keywords**

Keywords are words that occur in a corpus or a specific text more frequently than in a reference text. For example, a corpus of medical texts compared to a reference corpus would contain medical terms (medicines, illnesses, etc.), a corpus of linguistic research articles would contain linguistic terminology; a corpus of legal texts words like law, crime, article, etc.

In order to extract keywords of a corpus, one needs to extract the frequencies of all words in a specific corpus (these can be lemmata or wordforms, depending on what you want to find), and the frequency of all words in the reference corpus. After that, one can merge them in one frequency table and compute the **association measure**, which will represent the keyness of the words in the specific corpus. There are very many measures (with advantages and disadvantages). We will use the log-odds ratio, which is the easiest to compute in R. Other well-known measures are the chi-squared statistic, log-likelihood statistic, pointwise mutual information.

**Today we will focus on wordforms – strings of characters between white spaces. We won’t use the parser.**

Let us obtain the keywords in a small corpus of love songs. It is available online as *lovesongs.txt*. So, we need to take three steps:

1. Get the frequencies of the words in the corpus of love songs, as we did during the last session.

2. Get the frequencies of the words in the reference corpus. Here, we’ll use a text-only version of the Santa Barbara Corpus of Spoken American English. The corpus can be found here: <https://www.linguistics.ucsb.edu/research/santa-barbara-corpus>

3. Put all frequencies together and compute the keyness measure (log-odds ratio).

**1. Get the frequencies of all words in the corpus of love songs.**

1.1. Save the corpus locally. An example of love songs:

#Lana Del Rei. Love

Look at you kids with your vintage music

Comin' through satellites while cruisin'

You're part of the past, but now you're the future

Signals crossing can get confusing

It's enough just to make you feel crazy, crazy, crazy

Sometimes, it's enough just to make you feel crazy

You get ready, you get all dressed up

To go nowhere in particular

…

1.2. Open the corpus in R as lovesongs, each word is a separate string.

**lovesongs <- scan(file = file.choose(), what = "character", sep = " ", comment.char = "#", encoding = "UTF-8", quote = "")**

**#** Read 4496 items

**head(lovesongs)**

[1] "Look" "at" "you" "kids" "with" "your"

Note that the comments are gone!

1.3. Normalize the upper- and lowercase, remove punctuation and make a frequency list of strings:

**#convert to lowercase**

**lovesongs <- tolower(lovesongs)**

**head(lovesongs)**

[1] "look" "at" "you" "kids" "with" "your"

**#remove punctuation and empty strings**

**lovesongs <- gsub('[[:punct:] ]+', '', lovesongs)**

**lovesongs <- lovesongs[nchar(lovesongs)> 0]**

**#make a frequency list**

**lovesongs\_freq <- table(lovesongs)**

**#sort: the top 20 words**

**sort(lovesongs\_freq, decreasing = TRUE)[1:20]**

lovesongs

you i me the love and know it to on my in im

228 199 146 114 107 100 99 93 93 86 74 65 64

a like your we that be baby

63 63 58 56 51 45 41

1.4. Finally, we can make a data frame with the words as column 1, and the frequencies as column 2.

**lovesongs\_df <- data.frame(Word = names(lovesongs\_freq), Freq\_Love = as.numeric(lovesongs\_freq))**

**head(lovesongs\_df)**

Word Freq\_Love

1 a 63

2 about 4

3 abuse 1

4 act 1

5 acting 2

6 admit 7

**dim(lovesongs\_df)**

[1] 643 2 #That is, we have 643 words

**2. Get the frequencies in a reference corpus**

2.1. Open the corpus in R:

**sb <- scan(file = file.choose(), what = "character", sep = " ", comment.char = "#", encoding = "UTF-8", quote = "")**

**head(sb)**

**[1] "..." "So" "you" "don't" "need" "to"**

2.2. Clean the corpus, removing some expressions

The corpus contains paralinguistic information: GASP, HISS, SNEEZE, etc. We can remove all strings with 2 and more uppercase symbols:

**sb\_clean <- gsub("[A-Z]+{2,}", "", sb)**

Now we can clean up the punctuation and convert the symbols to lowercase, as we did with the love songs.

**sb****\_clean <- tolower(sb\_clean)**

**head(sb\_clean)**

[1] "..." "so" "you" "don't" "need" "to"

**#remove punctuation, numbers and empty strings**

**sb\_clean <- gsub('[[:punct:] ]+', '', sb\_clean)**

**head(sb\_clean)**

**[1] "" "so" "you" "dont" "need" "to"**

**sb\_clean <- gsub('[[:digit:] ]+', '', sb\_clean)**

**sb\_clean <- sb\_clean[nchar(sb\_clean)> 0]**

**head(sb\_clean)**

**[1] "so" "you" "dont" "need" "to" "go"**

**#We’ll also remove numerals:**

2.3. Make a frequency list

**sb\_freq <- table(sb\_clean)**

**sort(sb\_freq, decreasing = TRUE)[1:20]**

**sb\_clean**

**the i and you to a that it of in was know yeah**

**8324 8107 7858 6391 4878 4853 4587 4351 3272 2935 2854 2605 2549**

**he is like they so this uh**

**2392 2335 2286 2187 2151 2036 1893**

What are the differences with the lovesong corpus?

2.4. Finally, let’s make a data frame with the words as one column and the frequencies

**sb\_df <- data.frame(Word = names(sb\_freq), Freq\_Ref = as.numeric(sb\_freq))**

**head(sb\_df)**

**Word Freq\_Ref**

**1 a 4853**

**2 aa 1**

**3 aah 4**

**4 aaron 1**

**5 ab 1**

**6 abacubby 1**

**Step 3. Combine the frequencies and compute the keyness measure**

First, we combine the frequencies in one data frame:

**key\_df <- merge(sb\_df, lovesongs\_df, by = "Word", all = TRUE)**

**nrow(key\_df)**

[1] 11730 #total number of lemmata

**#We need to replace NAs with zeros**

**key\_df[is.na(key\_df)] <- 0**

**head(key\_df)**

**Word Freq\_Ref Freq\_Love**

**1 a 4853 63**

**2 aa 1 0**

**3 aah 4 0**

**4 aaron 1 0**

**5 ab 1 0**

**6 abacubby 1 0**

Our keyness measure is called **log-odds ratio**. The formula is as follows:

Keyness =log =

The small quantity 0.01 is added in order to avoid division by zero.

The measures Freq\_Love\_Other and Freq\_Ref\_Other are the frequencies of all other words (with the exception of the given word) in the corpus of interest.

**key\_df$Freq\_Love\_Other <- sum(key\_df$Freq\_Love) - key\_df$Freq\_Love**

**key\_df$Freq\_Ref\_Other <- sum(key\_df$Freq\_Ref) - key\_df$Freq\_Ref**

Finally, we can compute the keyness scores:

**key\_df$Keyness <- log((key\_df$Freq\_Love + 0.01)\*(key\_df$Freq\_Ref\_Other + 0.01)/((key\_df$Freq\_Ref + 0.01)\*(key\_df$Freq\_Love\_Other + 0.01)))**

Now we can examine the top 20 keywords in the love songs corpus. What are your conclusions?

**key\_df[order(-key\_df$Keyness),][1:20, c(1:3, 6)]**

Word Freq\_Ref Freq\_Love Keyness

11655 ahah 0 22 11.721290

11692 loving 0 16 11.401666

11683 heey 0 12 11.113299

11671 discovering 0 6 10.419647

11716 somethin 0 6 10.419647

11720 sunrise 0 6 10.419647

11697 niggas 0 5 10.237435

11712 skrrt 0 5 10.237435

11668 cursing 0 4 10.014568

11684 holdin 0 4 10.014568

11658 bedsheets 0 3 9.727494

11694 magnet 0 3 9.727494

11699 oh—i—oh—i—oh—i—oh—i 0 3 9.727494

11700 ohiohiohiohi 0 3 9.727494

11701 ooooohh 0 3 9.727494

11707 restless 0 3 9.727494

11713 slipping 0 3 9.727494

11715 smell 0 3 9.727494

11723 til 0 3 9.727494

11670 deny 0 2 9.323466

**Exercise**

Choose a music genre of your preference, compile a small corpus and compare it with the reference corpus.