**Introduction to Corpus Linguistics**

WiSe 2018-2019

by Dr. Natalia Levshina ©

**Session 3. 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. See Step 3.

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, e.g. UD corpus of English, which we have already used.

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 as we opened the UD corpora in Session 1, one line of text is a separate string.

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

**head(lovesongs)**

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

[2] "Comin' through satellites while cruisin'"

[3] "You're part of the past, but now you're the future"

[4] "Signals crossing can get confusing"

[5] "It's enough just to make you feel crazy, crazy, crazy"

[6] "Sometimes, it's enough just to make you feel crazy"

Note that the comments are gone!

1.3. Use the package udpipe to parse the corpus file:

**library(udpipe)**

**#Warning: the next commands are necessary only if you haven't #downloaded the model before. To check if you have the model, go to #the Files tab.**

**ud\_eng <- udpipe\_download\_model(language = "english")**

**ud\_eng <- udpipe\_load\_model(ud\_eng)**

**#If you have already downloaded the language model:**

**ud\_eng <- udpipe\_load\_model("english-ud-2.0-170801.udpipe") #change the name of the model if you work with another language!**

**#Now we can parse the corpus**

**lovesongs\_ud <- udpipe(x = lovesongs, object = ud\_eng)**

**head(lovesongs\_ud)**

**#(omitted)**

1.4. Make a frequency list of lemmas:

**#we need only lemmas for this case study. Alternatively, check #colnames(lovesongs\_ud)for the names of the other columns.**

**lovesongs\_lemmas <- lovesongs\_ud$lemma**

**#make a lemma frequency list**

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

**#sort: the top 20 words**

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

lovesongs\_lemmas

I , you be love the and it to get on my do

527 453 379 303 249 153 137 107 104 103 99 96 92

not that baby in a like (

91 78 75 74 73 72 65

**#note that we haven't got rid of punctuation here! It is not important here, but may be important for other purposes. Think about that first!**

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

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

**head(lovesongs\_df)**

Lemma Freq\_Love

1 ' 46

2 '97 1

3 'Cause 10

4 'em 12

5 'Long 6

6 's 6

**dim(lovesongs\_df)**

[1] 767 2 #That is, we have 767 lemmas

**2. Get the frequencies of all lemmas in the reference corpus**

2.1. Open the corpus in R:

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

**head(refcorpus)**

1] "1\tFrom\tfrom\tADP\tIN\t\_\t3\tcase\t3:case\t\_"

[2] "2\tthe\tthe\tDET\tDT\tDefinite=Def|PronType=Art\t3\tdet\t3:det\t\_"

[3] "3\tAP\tAP\tPROPN\tNNP\tNumber=Sing\t4\tobl\t4:obl:from\t\_"

**...**

2.2. Split each line by a tab and get all lemmas:

**ref\_list <- strsplit(refcorpus, "\t")**

**head(ref\_list)**

**#output omitted**

**#for each line (element in the list), get the lemma (element 3)**

**ref\_lemmas <- sapply(ref\_list, function (x) x[3])**

**head(ref\_lemmas)**

[1] "from" "the" "AP" "come" "this" "story"

2.3. Compute the frequencies of each lemma and examine the top 20:

**ref\_freq <- table(ref\_lemmas)**

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

ref\_lemmas

. be the , to and a I you of in have for

1140 987 982 800 565 561 557 432 394 391 366 330 241

it they not that do on ?

235 219 201 201 172 168 163

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

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

**head(ref\_df)**

Lemma Freq\_Ref

1 ' 19

2 's 85

3 - 150

4 -- 22

5 --- 1

6 ------------------------------------------------ 1

**dim(ref\_df)**

[1] 4391 2

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

First, we combine the frequencies in one data frame:

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

**nrow(key\_df)**

[1] 4693 #total number of lemmata

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

**key\_df[is.na(key\_df)] <- 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 keywords in the love songs corpus:

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

Lemma Freq\_Ref Freq\_Love Keyness

4408 baby 0 75 10.270889

4648 ta 0 36 9.531111

4501 harder 0 30 9.347932

4691 yeah 0 27 9.242153

4565 na 0 23 9.081266

4583 peat- 0 15 8.652840

4487 gon 0 13 8.509538

4394 'em 0 12 8.429408

4393 'Cause 0 10 8.246949

4402 alright 0 9 8.141548

4609 Sexy 0 9 8.141548

4400 ah 0 7 7.890248

4404 anymore 0 7 7.890248

4395 'Long 0 6 7.736183

4426 brand 0 6 7.736183

4514 i—oh— 0 6 7.736183

4515 I—oh 0 6 7.736183

4516 I—oh- 0 6 7.736183

4573 Oh- 0 6 7.736183

4637 sunrise 0 6 7.736183

**Exercise**

Choose a music genre of your preference, compile a small corpus and compare it with the reference corpus. It can be in English or in any other language that you know.