Unit #2: BNC Frequency Comparisons

Stefan Evert

28 November 2015

In this exercise, you will carry out various frequency comparisons for lexical and grammatical features between subcorpora of the British National Corpus. In doing so, you will also learn how to load, manipulate and export statistical tables, which are known as `data.frame`s in R lingo.

First, let us load some corpus frequency counts obtained with BNCweb (http://corpora.lancs.ac.uk/BNCweb/), which are stored as a text table in the file `datasets/bnc_queries.tbl`, using TAB characters as column delimiters (this format is particularly easy to process with R and many other tools). The file contains separate frequency counts for each text document in the British National Corpus, which allows us to compute frequencies for arbitrary subcorpora.

R has a built-in function `read.table` for loading statistical tables from text files; we use the variant `read.delim` with presets for TAB-delimited tables. Note the additional options, which you should always use (unless you have a good reason to do otherwise).

```
BNCqueries <- read.delim("bnc_queries.tbl", quote="", stringsAsFactors=FALSE)
```

If you are lazy, you can also use a pre-loaded version of this data frame available in the R package `SIGIL`.

```
library(SIGIL)
dim(BNCqueries)
```

```
## [1] 4048   12
```

The SIGIL package also contains a help page with further information about the data set, including a description of the *variables* (columns) in the table. You will also find queries in CEQL syntax that allow you to reproduce the frequency counts on a BNCweb server.

```
?BNCqueries
```

In order to compile interesting subcorpora, we also need metadata information for the individual text documents. A comprehensive metadata table is also provided in the form of a TAB-delimited text file (or pre-loaded in SIGIL as `BNCmeta`). Note that when dealing with language data, you should always specify the character encoding of the text file; otherwise accented letters and other special characters will not load correctly.

```
BNCmeta <- read.delim("bnc_metadata_utf8.tbl", quote="",
                      stringsAsFactors=FALSE, fileEncoding="utf8")
```

Read the help page `?BNCmeta` in order to find out which metadata are available for the different texts. In order to list the categories distinguished by a given metadat variable, it is easiest to tabulate the corresponding column of `BNCmeta`:

```
table(BNCmeta$derived_type) # What do those numbers mean?
```

```
##
##            academic             fiction      misc_published
##                 497                 452                 710
##           newspaper               prose spoken_conversation
##                 486                 744                 153
##        spoken_other         unpublished
##                 755                 251
```

We can also cross-tabulate different metadata variables to see how many texts there are for each combination. In order to avoid repeating `BNCmeta`, we can use the `with` function to address the columns of the table directly as variables:

```
with(BNCmeta, table(derived_type, author_sex))
```

```
##                     author_sex
## derived_type         --- female male mixed unknown
##   academic             0     41  240     5     211
##   fiction              0    212  218     1      21
##   misc_published       0     41  126    53     490
##   newspaper            0      0    0    95     391
##   prose                0    100  314    48     282
##   spoken_conversation 153      0    0     0       0
##   spoken_other        755      0    0     0       0
##   unpublished          0     20   22    32     177
```

Such tabulations show the number of texts in each category or combination of categories. Sometimes we would rather like to know how many word tokens or sentences there are. For this purpose, we need to add up the per-text token counts (`n_w`) or sentence counts (`n_s`), which can be done with the `xtabs` function (a more flexible variant of `table`). Note that `xtabs` uses the "*formula interface*" for specifying the cross-classifying factors and the dependent variable (i.e. the per-text frequencies).

```
xtabs(n_w ~ derived_type + author_sex, data=BNCmeta)
```

```
##                     author_sex
## derived_type             ---   female     male    mixed  unknown
##   academic                 0  1263207  8366916   185239  5962666
##   fiction                  0  8392300  7294780     1093   455740
##   misc_published           0  1270119  3454378  2120536 11079076
##   newspaper                0        0        0  1279223  8132951
##   prose                    0  3326936 11041452  1733995  8076291
##   spoken_conversation  4233962        0        0        0        0
##   spoken_other         6175896        0        0        0        0
##   unpublished              0   335692   504505  1218843  2407633
```

In order to compile a subcorpus of the BNC, we can apply the `subset` function to filter the rows of the metadata table according to certain criteria. While you can think of this as applying a condition to each row of the data frame, R actually uses vector operations for efficiency. So keep in mind that all logical operations have to be vectorised (i.e. use `&` rather than `&&`, and `|` rather than `||`).

```
FictM <- subset(BNCmeta, derived_type == "fiction" & author_sex == "male")
FictF <- subset(BNCmeta, derived_type == "fiction" & author_sex == "female")
```

Note that you can also use data frames to print multiple items of information in a tidy and compact way:

```
data.frame(male=nrow(FictM), female=nrow(FictF), row.names="# texts")
```

```
##         male female
## # texts  218    212
```

In order to obtain frequency counts for each subcorpus, we need to merge the frequency data from `BNCqueries` with the metadata information from `BNCmeta` (corresponding to a JOIN operation in a relational database). Such table joins can be computed with the `merge` function. Since we should not rely on the ordering of rows in the two tables, we specify that corresponding rows should be matched by the `id` variable (i.e. the BNC text ID). You can also specify multiple variables in case their is no unique ID that identifies rows unambiguously. All other variables from the two data frames will be combined into a single table.

```
BNC <- merge(BNCqueries, BNCmeta, by="id")
```

Read `?merge` to learn more about the different options of the function, which allow you to carry out many different subtypes of JOIN operations.

Now we can recompute the two subcorpora based on the combined table (you could also merge the data frames `FictM` and `FictF` with the relevant rows of `BNCqueries`, but it is more convenient to perform the JOIN just once).

```
FictM <- subset(BNC, derived_type == "fiction" & author_sex == "male")
FictF <- subset(BNC, derived_type == "fiction" & author_sex == "female")
```

If you prefer to view the subcorpora with an external spreadsheet program, you can save them in CSV (= comma-separated values) format, which is a standard text-based exchange format supported by most spreadsheets. Note that software with a German or French localization might use a different form of CSV with semicolons as separators (because commas are used as decimal points in these languages). Experiment with different CSV formats and character encodings to find a setting that's compatible with your favourite spreadsheet editor.

```
write.csv(FictM, file="fiction_male.csv", fileEncoding="utf8")
write.csv2(FictF, file="fiction_female.csv", fileEncoding="cp1252")
```

The last step is to compute the pooled frequency counts for each subcorpus, by adding up the relevant columns of the data frame. Let us look at the token frequency of downtoners such as *almost* or *scarcely* as an example:

```
k1 <- sum(FictM$downtoner)
n1 <- sum(FictM$n_w)
k2 <- sum(FictF$downtoner)
n2 <- sum(FictF$n_w)
```

Let us collect these data into a contingency table

```
ct <- cbind(c(k1, n1-k1), c(k2, n2-k2))
rownames(ct) <- c("downtoners", "other words") # optional
colnames(ct) <- c("male", "female")
ct
```

```
##                male  female
## downtoners    16392   20435
## other words 7278388 8371865
```

Now we can apply various hypothesis tests for contingency tables

```
chisq.test(ct)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  ct
## X-squared = 58.74, df = 1, p-value = 1.799e-14
```

```
fisher.test(ct)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  ct
## p-value = 1.677e-14
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.9038133 0.9418732
## sample estimates:
## odds ratio
##  0.9226392
```

The frequency difference is highly significant. Fisher's exact test also returns a confidence interval for the odds ratio θ as a measure of effect size. Can you interpret this value? Is the difference linguistically *meaningful*?

A more intuitive effect size measure, the difference of proportions δ, can be computed with the specialized proportions test. Keep in mind that the `prop.test` expects a different input format than a contingency table.

```
prop.test(c(k1, k2), c(n1, n2))
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  c(k1, k2) out of c(n1, n2)
## X-squared = 58.74, df = 1, p-value = 1.799e-14
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.0002358923 -0.0001398754
## sample estimates:
##      prop 1      prop 2
## 0.002247086 0.002434970
```

Can you interpret this confidence interval? What is the difference between men and women?

Now it's your time to experiment with other subcorpora and queries. Can you find interesting expected or unexpected differences? How do you compute n for frequencies given as sentence counts? And what is a reasonable sample size for a comparison of the frequency of split infinitives (`split.inf.S`)?

Advanced R users should also try to automate these comparisons, either by defining their own functions or by using other data manipulation and summation tools (so it is not necessary to generate a separate table for each subcorpus).