Over-represented words: test cases

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#data <- read.table("/Users/transfer/Downloads/Over-represented words test cases - Sheet1.csv", sep = "
#head(data)


# IDs of the elements in the arrays

names <- c("Corpus ID", "Size in tokens", "Smoothing factor", "Z",  "E",    "D",    "C",    "B",    "A"
# first element: the corpus ID (1 or 2)
# second element: the total size of the corpus, in tokens
# all other elements: count in tokens of each word in the vocabulary
# NB: these should be of equal length.
corpus.of.interest <- as.integer(c(1,   1406,   100,    5,  500,    400,    300,    200,    1,  0,  0,
corpus.reference <- as.integer(c(2, 906,    100,    5,  0,  0,  0,  0,  1,  400,    300,    200,    100
corpus.of.interest
```

```
##  [1]    1 1406  100    5  500  400  300  200    1    0    0    0    0
```

```
smoothing.factor <- 100
smoothing.factor
```

```
## [1] 100
```

```
corpus.of.interest.smoothed <- corpus.of.interest + smoothing.factor
corpus.of.interest.smoothed
```

```
##  [1]  101 1506  200  105  600  500  400  300  101  100  100  100  100
```

```
corpus.reference.smoothed <- corpus.reference + smoothing.factor

# subtract 2 from the length of the array because the first element is the ID of the corpus and the sec
corpus.of.interest.adjusted.corpus.size <-
corpus.of.interest[2] + (length(corpus.of.interest) - 2) * smoothing.factor
corpus.of.interest.adjusted.corpus.size
```

```
## [1] 2506
```

```
corpus.reference.adjusted.corpus.size <-
corpus.reference[2] + (length(corpus.reference) - 2) * smoothing.factor

# NB: the two relative.frequencies arrays are now 2 elements shorter than the original arrays
corpus.of.interest.relative.frequencies <- corpus.of.interest.smoothed[4:length(corpus.of.interest)] /
corpus.of.interest.relative.frequencies
```

```
##  [1] 0.04189944 0.23942538 0.19952115 0.15961692 0.11971269 0.04030327
##  [7] 0.03990423 0.03990423 0.03990423 0.03990423
```

```
corpus.reference.relative.frequencies <- corpus.reference.smoothed[4:length(corpus.reference)] / corpus

# We no longer ignore the first two, because the relative.frequencies arrays only contain counts of the
#ratios <- corpus.of.interest.relative.frequencies[3:length(corpus.of.interest)] / corpus.reference.rel
ratios <- corpus.of.interest.relative.frequencies / corpus.reference.relative.frequencies

ratios
```

```
##  [1] 0.8004789 4.8028731 4.0023943 3.2019154 2.4014366 0.8004789 0.1600958
##  [8] 0.2001197 0.2668263 0.4002394
```

```
#print(paste("Ratios should be:"), ratios)

names.vocabulary <- names[4:length(names)]
#print(names, ratios)
for (i in 1:length(names.vocabulary)) {
  print(paste(names.vocabulary[i], ratios[i]))
}
```

```
## [1] "Z 0.80047885075818"
## [1] "E 4.80287310454908"
## [1] "D 4.0023942537909"
## [1] "C 3.20191540303272"
## [1] "B 2.40143655227454"
## [1] "A 0.80047885075818"
## [1] "Y 0.160095770151636"
## [1] "X 0.200119712689545"
## [1] "W 0.26682628358606"
## [1] "V 0.40023942537909"
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.