Data Science capstone - Milestone Report (3% samples)

Anatoli Kraev

12/19/2019

Coursera Data Science Capstone Week 2 Milestone Report (adittion)

This paper discusses building n-gram for a 3% sample from the databases presented for capstone.

```
library(quanteda)
```

```
## Package version: 1.5.2
```

```
## Parallel computing: 2 of 6 threads used.
```

```
## See https://quanteda.io for tutorials and examples.
```

```
##
## Attaching package: 'quanteda'
```

```
## The following object is masked from 'package:utils':
##
##     View
```

Set the number of information processing threads

```
quanteda_options("threads" = 6)
```

Import data

```
pathEnTwit <- "final/en_US/en_US.twitter.txt"
pathEnNews <- "final/en_US/en_US.news.txt"
pathEnBlog <- "final/en_US/en_US.blogs.txt"
enBlogs <- readLines(pathEnBlog, warn = F, encoding = "UTF-8")
enNews <-  readLines(pathEnNews, warn = F, encoding = "UTF-8")
enTwit <- readLines(pathEnTwit, warn = F, encoding = "UTF-8")
corpEnBlogs <- corpus(enBlogs)
corpEnNews <- corpus(enNews)
corpEnTwit <- corpus(enTwit)
rm(enBlogs, enNews, enTwit)
```

Create the 3% sample:

```
set.seed(1234)
corpMinBlog <- corpus_sample(corpEnBlogs, 27000)
corpMinNews <- corpus_sample(corpEnNews, 30300)
corpMinTwit <- corpus_sample(corpEnTwit, 70800)
```

Cleaning data and creating n-gram databases for each database (news, blogs, twitter)

When creating n-gram databases, only the sequential arrangement of words without their omissions was used. For example, the phrase: “I love you” gives two n-grams: I_love and love_you.

For news data:

```
tokenEnNews <- tokens(corpMinNews, remove_punct = TRUE)
tokenEnNews <- tokens_remove(tokenEnNews, pattern = stopwords('en
'))

ngramNews <- tokens_ngrams(tokenEnNews, n = 1)
topNgramNews <- topfeatures(dfm(ngramNews), 40)
rm(ngramNews)
bigramNews <- tokens_ngrams(tokenEnNews, n = 2)
topBigramNews <- topfeatures(dfm(bigramNews), 40)
rm(bigramNews,tokenEnNews)
```

For blogs data:

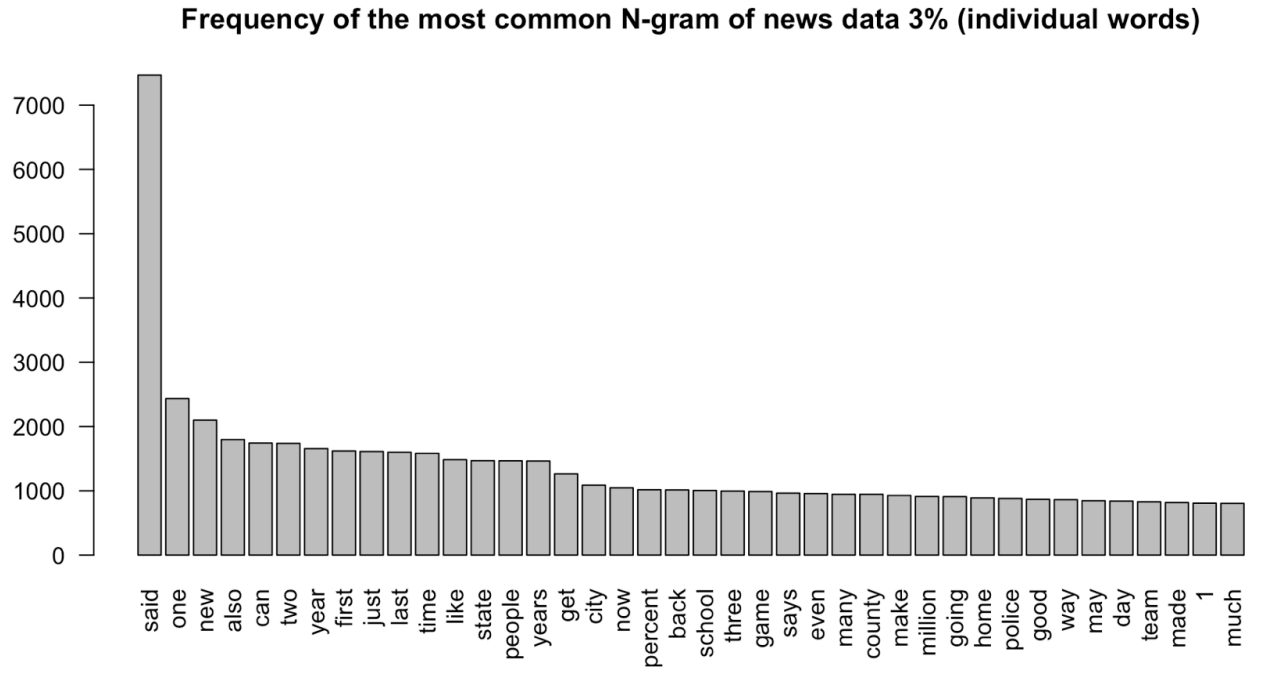
```
tokenBlog <- tokens(corpMinBlog, remove_punct = TRUE)
tokenBlog <- tokens_remove(tokenBlog, pattern = stopwords('en'))

ngramBlogs <- tokens_ngrams(tokenBlog, n = 1)
topNgramBlogs <- topfeatures(dfm(ngramBlogs), 40)
rm(ngramBlogs)
bigramBlogs <- tokens_ngrams(tokenBlog, n = 2)
topBigramBlogs <- topfeatures(dfm(bigramBlogs), 40)
rm(bigramBlogs, tokenBlog)
```

For twitter data:

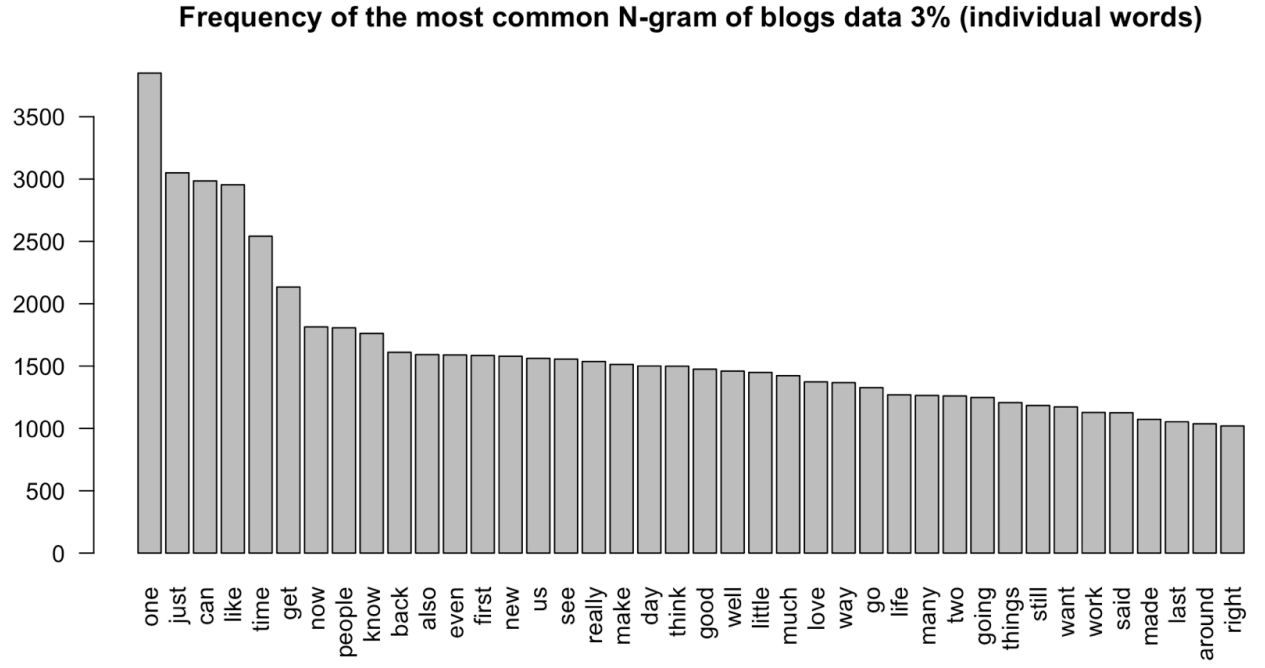
```
tokenTwit <- tokens(corpMinTwit, remove_punct = TRUE)
tokenTwit <- tokens_remove(tokenTwit, pattern = stopwords('en'))

ngramTwits <- tokens_ngrams(tokenTwit, n = 1)
topNgramTwits <- topfeatures(dfm(ngramTwits), 50)
rm(ngramTwits)
bigramTwits <- tokens_ngrams(tokenTwit, n = 2)
topBigramTwits <- topfeatures(dfm(bigramTwits), 50)
rm(bigramTwits, tokenTwit)
```

Frequency of the most common n-gram in databases (3% sample):

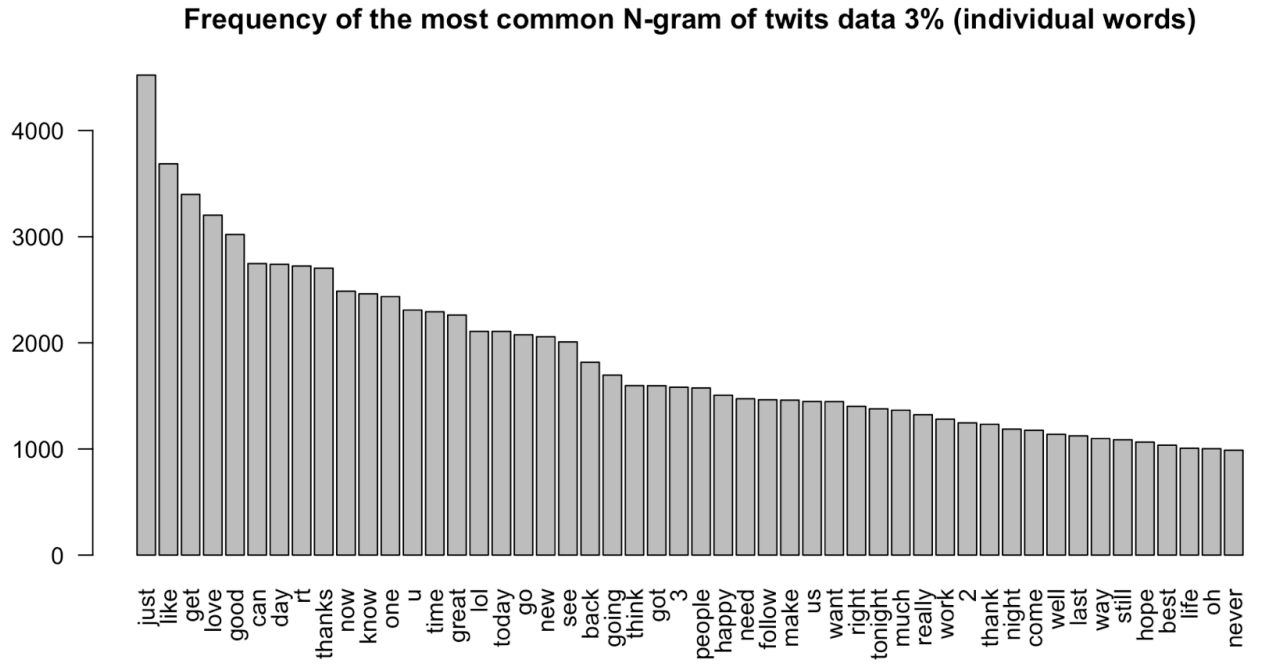
```
par(mar=c(4,4,4,4))
barplot(height = topNgramNews, names.arg = names(topNgramNews),
 las = 2, main = "Frequency of the most common N-gram of news dat
a 3% (individual words)")
```




```
barplot(height = topNgramBlogs, names.arg = names(topNgramBlogs),
        las = 2, main = "Frequency of the most common N-gram of b
logs data 3% (individual words)")
```

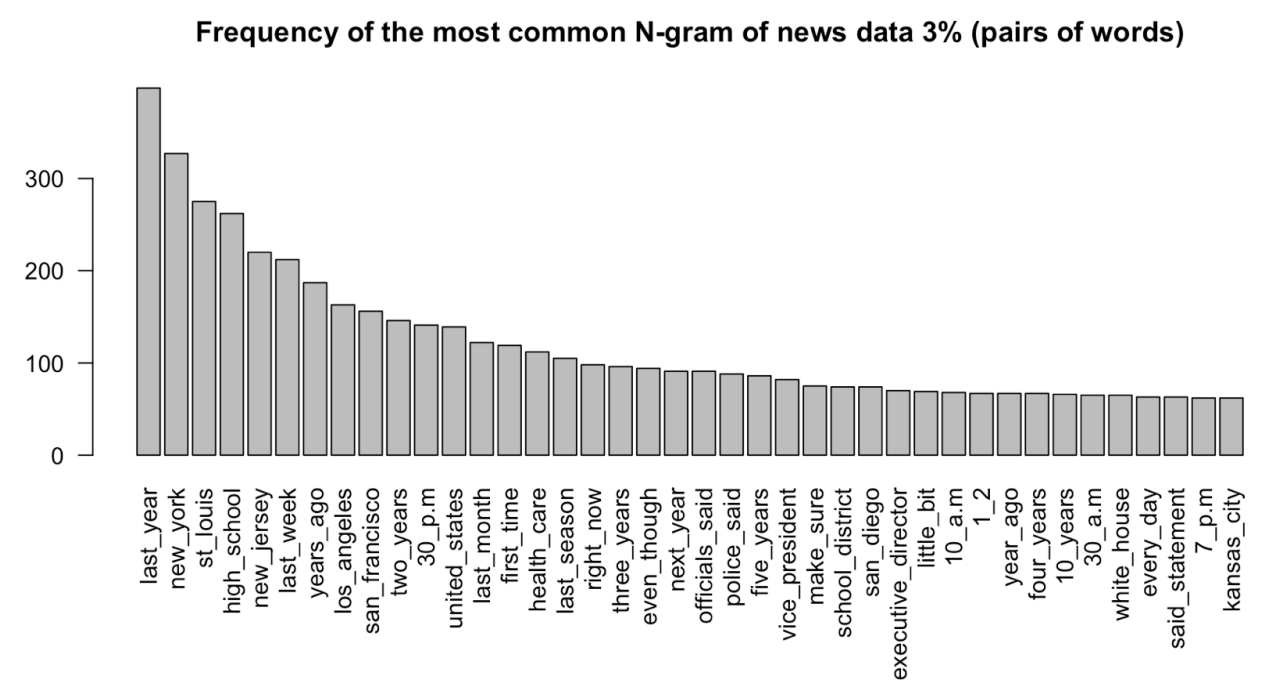



```
barplot(height = topNgramTwits, names.arg = names(topNgramTwits),
        las = 2, main = "Frequency of the most common N-gram of t
wits data 3% (individual words)")
```

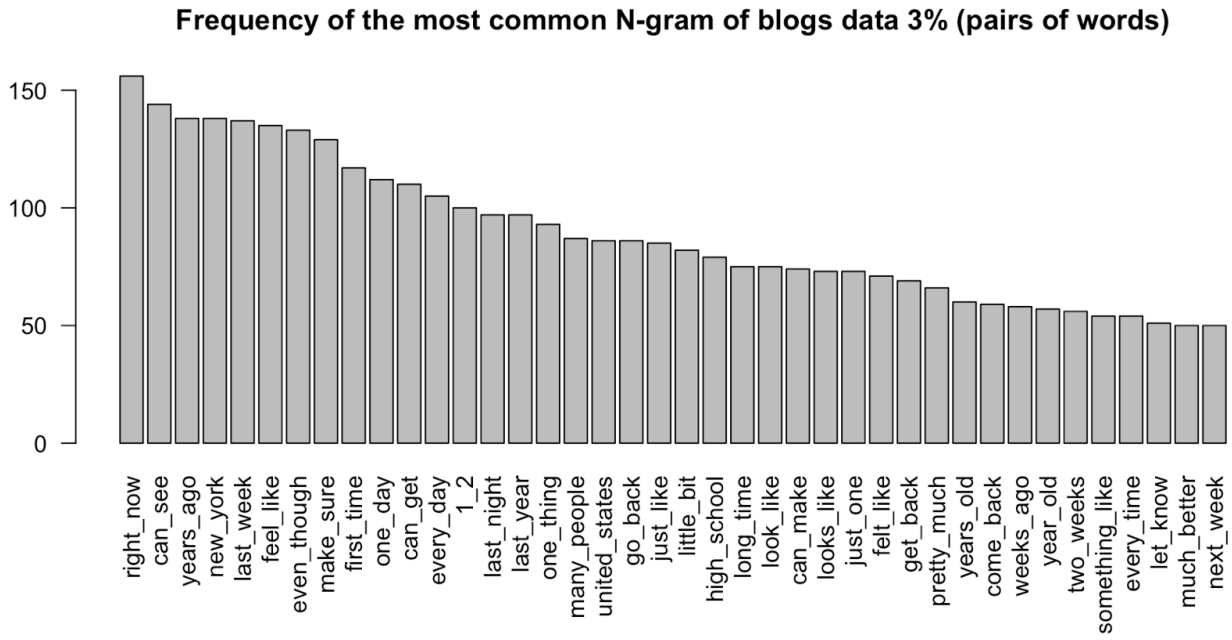


Frequency of the most common bi n-gram in databases (3% sample):

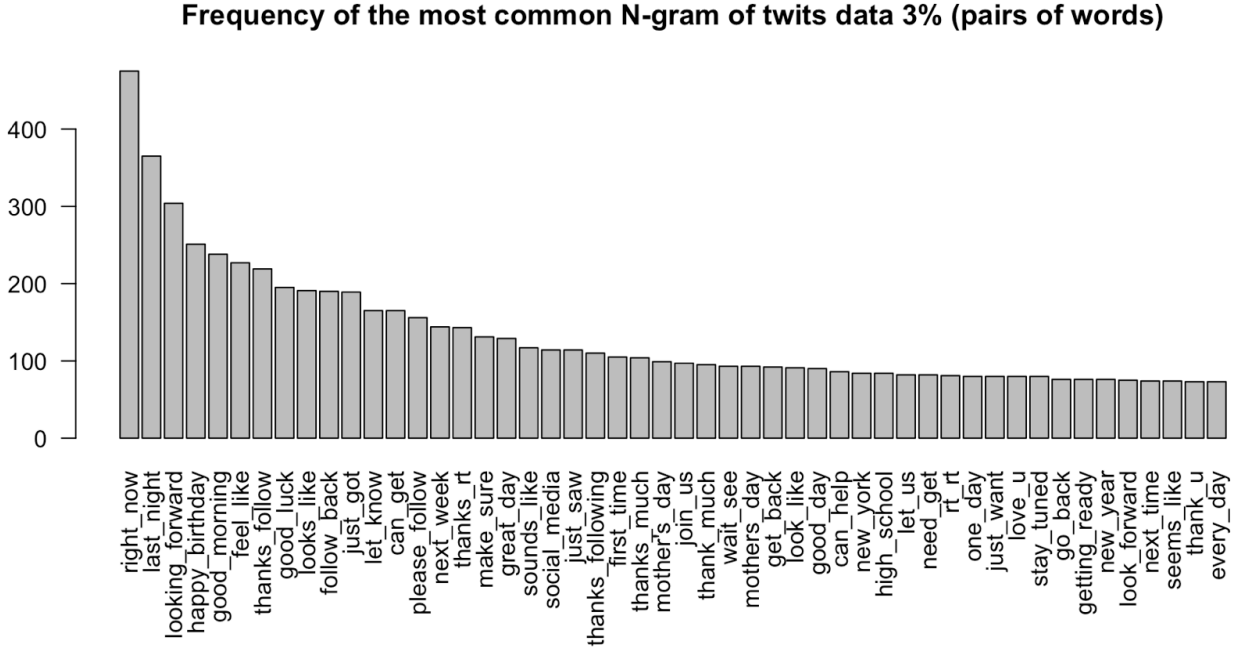
```
par(mar=c(8,4,4,4))
barplot(height = topBigramNews, names.arg = names(topBigramNews),
        las = 2, main = "Frequency of the most common N-gram of n
ews data 3% (pairs of words)")
```



```
barplot(height = topBigramBlogs, names.arg = names(topBigramBlogs
),
        las = 2, main = "Frequency of the most common N-gram of b
logs data 3% (pairs of words)")
```




```
barplot(height = topBigramTwits, names.arg = names(topBigramTwits
),
        las = 2, main = "Frequency of the most common N-gram of t
wits data 3% (pairs of words)")
```



Summary

Comparing barcharts for the main databases and 3% of the samples, it can be noted that n-grams barcharts have a very good match. The bi-n-grams barcharts are mostly the same, while the order of some of them has changed. In general, 3% of the sample repeats the properties of the main databases.