---

title: "Capstone: Milestone Report" output: html_document

March 15, 2016

Overview

This is the milestone report for the Data Science Capstone Project.

---

title: "Capstone: Milestone Report" output: html_document

The goal of this report was to build a simple model for the relationship between words, as a first step in creating a predictive text mining application.

---

title: “Capstone: Milestone Report” output: html_document

The following sections describe my methods for analysing the datasets

Libraries

Load the necessary libraries

```
r
library(tm)
library(knitr)
```

---

title: "Capstone: Milestone Report" output: html_document

Securing the Data and Preliminary Analyses

The dataset can be downloaded from here

* https://d396qusza40orc.cloudfront.net/dsscapstone/dataset/Coursera-SwiftKey.zip

---

title: “Capstone: Milestone Report” output: html_document

Included are three different data files containing text sampled from blogs, news articles, and twitter feeds. The the English versions was used.

---
title: "Capstone: Milestone Report" output: html_document

```
r ## Read in the files blog <- readLines("data/final/en_US/en_US.blogs.txt") news <- readLines("data/final/en_US/en_US.news.txt") twitter <- readLines("data/final/en_US/en_US.twitter.txt")
```

Combined the datasets contain several million lines of text, with over 20 million characters.

---

title: "Capstone: Milestone Report" output: html_document

```
r
kable(data.frame( "Data File" = c("Blogs", "News", "Twitter"), "Line Count" = c(length(blog), length(news), length(twitter)), "Character Count" = c( sum(nchar(blog)), sum(nchar(news)), sum(nchar(twitter))) ))
```

Data.File	Line.Count	Character.Count
Blogs	899288	208361438
News	77259	15683765
Twitter	2360148	162384825

Clean and sample the data sets

In order to be able to predict the next word with the highest degree of accuracy, in a reasonably efficient manner, the data set needed to be cleaned up. Numbers, punctuation, special characters, and stop words were removed. In addition, words where converted back to their stems.

The next sections uses the tm package, you can find an introduction here * https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf

Samples of the data sets are used to reduce the memory footprint.

```r
set.seed(90210)

combined_set <- c(blog, news, twitter)
sample_set   <- sample(combined_set, size = 3000, replace = TRUE)

# encode
Encoding(sample_set) <- "latin1"
combined <- iconv(sample_set, "latin1", "ASCII", "")

# create the corpus
swiftkey = Corpus(VectorSource(sample_set))

# clean up objects
#rm(blog, news, twitter, sample_set)
```

Tidy up the data sets by removing elements and converting to a common case.

```r
swiftkey <- tm_map(swiftkey, content_transformer(tolower))

# remove the most commonly used words in the english language
swiftkey <- tm_map(swiftkey, removeWords, stopwords("english"))

# reduce inflected (or sometimes derived) words to their word stem
swiftkey <- tm_map(swiftkey, stemDocument)

# clean up
swiftkey <- tm_map(swiftkey, stripWhitespace)
swiftkey <- tm_map(swiftkey, removePunctuation)
swiftkey <- tm_map(swiftkey, removeNumbers)
```

Basic n-gram models

```r
ff <- TermDocumentMatrix(swiftkey)
findFreqTerms(ff, lowfreq=100)
```

```
##  [1] "back"  "can"   "come"  "day"   "get"   "good"  "just"  "know"
##  [9] "like"  "look"  "love"  "make"  "need"  "new"   "now"   "one"
## [17] "see"   "thank" "think" "time"  "want"  "will"
```

end with word cloud for fun