Data Preparation & Cleaning

```
library(readxl)
library(dplyr)
library(magrittr)
library(tm)
library(keras)
```

```
## Warning: package 'keras' was built under R version 3.5.3
```

```
library(SnowballC)
```

1. Loading The Data

We will only load the first 100 observations

```
# Encode specific kinds of values as NA while reading excel
non_bill_df <- read_excel("data/december_non-bill_calls.xlsx", na = c("", "---"), n_max = 1000)
billed_df <- read_excel("data/december_billed_calls.xlsx", na = c("", "---"), n_max = 1000)
```

We will combine `non_bill_df` and `billed_df` into a dataframe called `billing_df` .

```
billing_df <- bind_rows(non_bill_df, billed_df)

str(billing_df, nchar.max = 20, vec.len = 3)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    2000 obs. of  29 variables:
##  $ Invoiced (Y/N)                      : chr  "N" "N" "N" ...
##  $ SR Number                           : chr  "1-ICUU0R5" "1-IR7JD4O" "1-IYEXL4A" ...
##  $ Activity Facts Call Num             : num  5.92e| __truncated__ ...
##  $ Br Branch Desc                      : chr  "114"| __truncated__ "114"| __truncated__ "115"|
__truncated__ ...
##  $ SR Owner (Q#)                       : chr  "FS5-0000000034" "FS5-0000000010" "FS5-000000001
0" ...
##  $ Billing Notes                       : chr  NA NA "1. "| __truncated__ ...
##  $ Call Text                           : chr  "ATM"| __truncated__ "DEC"| __truncated__ "DEC"|
__truncated__ ...
##  $ Cash Vendor & Consumable Contracts: logi  NA NA NA NA ...
##  $ SR Type                             : chr  "TR - Trouble Call" "TR - Trouble Call" "TR - Tro
uble Call" ...
##  $ Coverage Type                       : chr  NA "PL "| __truncated__ "CP "| __truncated__ ...
##  $ SR Coverage Hours                   : chr  NA "FK" "FK" ...
##  $ SR Device                           : chr  "VLT" "VAT" "VLT" ...
##  $ Item Desc                           : chr  NA "CVN"| __truncated__ "VAU"| __truncated__ ...
##  $ Activity Trouble Code               : chr  "REPAIR" "REPAIR" "PS_PRODUCT SALE" ...
##  $ SR Address Line 1                   : chr  "CAS"| __truncated__ "4510 KINGWOOD DR" "350"| __
truncated__ ...
##  $ SR City                             : chr  "Cumming" "KINGWOOD" "RALEIGH" ...
##  $ SR State                            : chr  "GA" "TX" "NC" ...
##  $ SR Site                             : num  40482| __truncated__ ...
##  $ SR Serial Number                    : chr  NA "930506000074" "30321003411" ...
##  $ SR Contact Date                     : POSIXct, format: "201"| __truncated__ "201"| __truncat
ed__ ...
##  $ Activity Completed Date             : POSIXct, format: "201"| __truncated__ "201"| __truncat
ed__ ...
##  $ Br Region Desc                      : chr  "P24"| __truncated__ "P24"| __truncated__ "P24"|
__truncated__ ...
##  $ Br Area Desc                        : chr  "P24"| __truncated__ "P24"| __truncated__ "P24"|
__truncated__ ...
##  $ Activity Type                       : chr  "Field Repair" "Field Repair" "Field Repair" ...
##  $ SR Status                           : chr  "Closed" "Closed" "Closed" ...
##  $ Activity Status                     : chr  "Closed Complete" "Closed Complete" "Closed Compl
ete" ...
##  $ Charges Status                      : chr  "Final" "Final" "Final" ...
##  $ SR Coverage Hours__1                : chr  NA "FK" "FK" ...
##  $ Base Call YN                        : chr  "Y" "Y" "Y" ...
```

Based on initial discussions and research into the meaning of some of the features in this dataset, we have categorized the following features as being not **important**.

```
## [1] "SR Address Line 1"       "SR City"
## [3] "SR Status"               "Activity Status"
## [5] "Charges Status"          "SR Coverage Hours"
## [7] "SR Coverage Hours"       "Br Region Desc"
## [9] "Activity Facts Call Num"
```

The features SR Coverage Hours…11 and SR Coverage Hours…28 were created by R because the excel contained two columns with the name SR Coverage Hours.

The features have been stored in variable called features_to_rm, the next step is to remove these r length(features_to_rm) features from the billing_df dataset. This step reduces our number of features from **29 to 20**.

```
billing_df <- billing_df %>% select(-features_to_rm)

str(billing_df, nchar.max = 20, vec.len = 3)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    2000 obs. of  21 variables:
##  $ Invoiced (Y/N)                    : chr  "N" "N" "N" ...
##  $ SR Number                         : chr  "1-ICUU0R5" "1-IR7JD4O" "1-IYEXL4A" ...
##  $ Br Branch Desc                    : chr  "114"| __truncated__ "114"| __truncated__ "115"|
__truncated__ ...
##  $ SR Owner (Q#)                     : chr  "FS5-0000000034" "FS5-0000000010" "FS5-000000001
0" ...
##  $ Billing Notes                     : chr  NA NA "1. "| __truncated__ ...
##  $ Call Text                         : chr  "ATM"| __truncated__ "DEC"| __truncated__ "DEC"|
__truncated__ ...
##  $ Cash Vendor & Consumable Contracts: logi  NA NA NA NA ...
##  $ SR Type                           : chr  "TR - Trouble Call" "TR - Trouble Call" "TR - Tro
uble Call" ...
##  $ Coverage Type                     : chr  NA "PL "| __truncated__ "CP "| __truncated__ ...
##  $ SR Device                         : chr  "VLT" "VAT" "VLT" ...
##  $ Item Desc                         : chr  NA "CVN"| __truncated__ "VAU"| __truncated__ ...
##  $ Activity Trouble Code             : chr  "REPAIR" "REPAIR" "PS_PRODUCT SALE" ...
##  $ SR State                          : chr  "GA" "TX" "NC" ...
##  $ SR Site                           : num  40482| __truncated__ ...
##  $ SR Serial Number                  : chr  NA "930506000074" "30321003411" ...
##  $ SR Contact Date                   : POSIXct, format: "201"| __truncated__ "201"| __truncat
ed__ ...
##  $ Activity Completed Date           : POSIXct, format: "201"| __truncated__ "201"| __truncat
ed__ ...
##  $ Br Area Desc                      : chr  "P24"| __truncated__ "P24"| __truncated__ "P24"|
__truncated__ ...
##  $ Activity Type                     : chr  "Field Repair" "Field Repair" "Field Repair" ...
##  $ SR Coverage Hours__1              : chr  NA "FK" "FK" ...
##  $ Base Call YN                      : chr  "Y" "Y" "Y" ...
```

2. Cleaning The Data

2.1 Encoding The Variables

We can notice that R has miss categorized some of the features in our dataset. There are certain features that are supposed to be read as categorical such as:

```
##  [1] "Invoiced (Y/N)"
##  [2] "Activity Type"
##  [3] "Activity Trouble Code"
##  [4] "Coverage Type"
##  [5] "Base Call YN"
##  [6] "SR Type"
##  [7] "SR Device"
##  [8] "SR Site"
##  [9] "SR Owner (Q#)"
## [10] "SR Serial Number"
## [11] "Cash Vendor & Consumable Contracts"
```

Lets encode the features in `char_to_factors` as factors

```
billing_df <- billing_df %>% mutate_at(char_to_factors, factor)

billing_df %>% summary()
```

```
##  Invoiced (Y/N)  SR Number         Br Branch Desc           SR Owner (Q#)
##  N:1000         Length:2000        Length:2000        FS5-0000000008:815
##  Y:1000         Class :character   Class :character   FS5-0000000010:574
##                 Mode  :character   Mode  :character   FS5-0000000023:347
##                                                       FS5-0000000034: 79
##                                                       FS5-REGIONAL  : 40
##                                                       FS5-0000000017: 32
##                                                       (Other)       :113
##  Billing Notes       Call Text         Cash Vendor & Consumable Contracts
##  Length:2000        Length:2000        NA's:2000
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
##                  SR Type
##  FL - First Line Call:693
##  IN - BW             :608
##  PM/CL               :  5
##  TR - Trouble Call   :694
##
##
##
##                                              Coverage Type SR Device
##  CP PLAN:  Parts and Labor                            :352   MSC:  11
##  GO PLAN:  Gold Coverage - Conventional & ATM Produ:202   TAB:1490
##  FIRST LINE (WINCOR CINEO 2560DA) - MANAGED PLAN   :162   VAT: 103
##  FIRST LINE (WINCOR CINEO 2590DA) - MANAGED PLAN   :139   VLT: 396
##  PL PLAN:  Platinum Coverage - Conventional & ATM P: 70
##  (Other)                                           :389
##  NA's                                              :686
##   Item Desc                   Activity Trouble Code   SR State
##  Length:2000        REPAIR              :536       Length:2000
##  Class :character   INSTALLATION        :455       Class :character
##  Mode  :character   XX_NONE OF THE ABOVE:228       Mode  :character
##                     FL_FIRSTLINE        :198
##                     JM_JAMMED           :137
##                     DRILLING_FORCED     :126
##                     (Other)             :320
##     SR Site         SR Serial Number SR Contact Date
##  214304 :   8   891218000098:   7    Min.   :2018-01-23 02:53:00
##  1846360:   6   613001012   :   6    1st Qu.:2018-10-23 08:00:00
##  558288 :   5   10628001804 :   5    Median :2018-10-23 08:00:00
##  1849420:   4   418000278   :   4    Mean   :2018-10-26 03:14:52
##  2007375:   4   5300730402  :   4    3rd Qu.:2018-12-01 08:32:00
##  3848711:   4   (Other)     :1327    Max.   :2018-12-02 14:38:00
##  (Other):1969   NA's        : 647
##  Activity Completed Date       Br Area Desc                 Activity Type
##  Min.   :2018-12-01 01:00:00   Length:2000        Cleaning Only    :  5
##  1st Qu.:2018-12-02 14:48:00   Class :character   Field Repair     :688
##  Median :2018-12-07 13:09:30   Mode  :character   First Line       :693
##  Mean   :2018-12-11 23:00:47                      Forced Entry     :  5
##  3rd Qu.:2018-12-20 11:09:15                      Installation - BW:608
##  3rd Qu.:2018-12-20 11:09:15                     Installation
##  Max.   :2018-12-31 20:30:00                     Job Training     :  1
##
##  SR Coverage Hours__1 Base Call YN
##  Length:2000          Y:2000
##  Class :character
##  Mode  :character
##
##
##
##
```

2.2 Free Form Text

The features in our dataset that are free form text are the features `Billing Notes` and `Call Text`.

Below is a preview of `Call Text`

```
billing_df$`Call Text` %>% head(3)
```

```
## [1] "ATM head came in damaged in shipping . head is not fixable . lead time on parts 2-3 mont
hs to replace AHD Head ordered 00016760000C night drop head that was damaged . job completed"
## [2] "DECAL Generated Call, Customer Product ID 9912V. SEC6*DRIVE THRU*EQUIPMENT - NEW OR REPL
ACE*Please proceed with the proposal to order and install (2) new headsets.  NTE:$1144  Turned o
ver to Ron Freeman. Proposal submitted to CPG on 9/26/2018. Pending approval. Proposal decline
d."
## [3] "DECAL Generated Call, Customer Product ID 9101L. SEC6*TELLER UNDERCOUNTER STEEL*EQUIPMEN
T - NEW*Please  Portable Teller Locker and a Teller bus per proposal.NTE:$2161  09/20/18  1:16:2
8PM - EDT  Michael  per tech threatt, assign to tech kennedy. wjy Per PM Casey Hill. schedule fo
r the evening of 12/4/18. Per PM Casey Hill work completed on 12/4/18. Closing call complete."
```

Below is a preview of `Billing Notes`

```
billing_df$`Billing Notes` %>% extract(c(3, 5, 1))
```

```
## [1] "1. PLEASE PROVIDE REPAIR TIME FOR BILLING. THANKS. 2. SENDING TO SPECIAL QUEUE FOR ACCOU
NT TEAM REVIEW 3. NON-BILL. WILL BE BILLED ON SO"
## [2] "1. ON WHICH PIECE OF EQUIPMENT COMBOS CHANGED?  AND OK TO BILL FOR KEYS. 2. NO BILL SR C
ONTRACT WORK."
## [3] NA
```

```
call_text <-  use_series(billing_df, `Call Text`)
billing_notes <-  use_series(billing_df, `Billing Notes`)
item_desc <- use_series(billing_df, `Item Desc`)
```

```
call_text_corpus <- VCorpus(VectorSource(call_text), readerControl = list(language = "en"))
bill_notes_corpus <- VCorpus(VectorSource(billing_notes), readerControl = list(language = "en"))
item_desc_corpus <- VCorpus(VectorSource(item_desc), readerControl = list(language = "en"))
```

```
call_text_corpus %>% extract(1:3) %>% inspect()
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 3
##
## [[1]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 180
##
## [[2]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 280
##
## [[3]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 373
```

```
call_text_corpus %>% head(3) %>% lapply(function (doc) doc$content)
```

```
## $`1`
## [1] "ATM head came in damaged in shipping . head is not fixable . lead time on parts 2-3 mont
hs to replace AHD Head ordered 00016760000C night drop head that was damaged . job completed"
##
## $`2`
## [1] "DECAL Generated Call, Customer Product ID 9912V. SEC6*DRIVE THRU*EQUIPMENT - NEW OR REPL
ACE*Please proceed with the proposal to order and install (2) new headsets.  NTE:$1144  Turned o
ver to Ron Freeman. Proposal submitted to CPG on 9/26/2018. Pending approval. Proposal decline
d."
##
## $`3`
## [1] "DECAL Generated Call, Customer Product ID 9101L. SEC6*TELLER UNDERCOUNTER STEEL*EQUIPMEN
T - NEW*Please  Portable Teller Locker and a Teller bus per proposal.NTE:$2161  09/20/18  1:16:2
8PM - EDT  Michael  per tech threatt, assign to tech kennedy. wjy Per PM Casey Hill. schedule fo
r the evening of 12/4/18. Per PM Casey Hill work completed on 12/4/18. Closing call complete."
```

To clean our data set we will have to:

- Convert the text to lower case, so that words like “write” and “Write” are considered the same word
- Remove numbers
- Remove English stopwords e.g “the”, “is”, “of”, etc.
- Remove punctuation e.g “,”, “?”, etc.
- Eliminate extra white spaces
- Stemming our text

Using the `tm` package we will apply transformations to each text document in the `call_text_corpus` to clean the text document.

```
replace_asterix <- function(document) {
  gsub(pattern = "\\*", replacement = " ", document)
}

add_space_period <- function(document) {
  gsub(pattern = "\\.", replacement = ". ", document)
}

remove_single_chars <- function(document) {
  gsub(pattern = "\\s[a-z]\\s", replacement = " ", document)
}

clean_up <- function(corpus) {
  corpus %>%
    # Convert the text to lower case
    tm_map(content_transformer(tolower)) %>%
    # Replace asterics "*" with an empty space
    tm_map(content_transformer(replace_asterix)) %>%
    # Add a space after a period
    tm_map(content_transformer(add_space_period)) %>%
    # Remove numbers
    tm_map(removeNumbers) %>%
    # Remove english common stopwords
    tm_map(removeWords, stopwords("english")) %>%
    # Remove words related to time
    tm_map(removeWords, c("pm", "am", "edt")) %>%
    # Remove punctuations
    tm_map(removePunctuation) %>%
    # Remove orphaned letters
    tm_map(content_transformer(remove_single_chars)) %>%
    # Eliminate extra white spaces
    tm_map(stripWhitespace) %>%
    # strip trailing and leading whitespace
    tm_map(content_transformer(trimws)) %>%
    # Stem words
    tm_map(stemDocument)
}

call_text_cleaned <- clean_up(call_text_corpus)
bill_notes_cleaned <- clean_up(bill_notes_corpus)
item_desc_cleaned <- clean_up(item_desc_corpus)
```

```
call_text_cleaned  %>% lapply(function (doc) doc$content) %>% extract(1:5)
```

```
## $`1`
## [1] "atm head came damag ship head fixabl lead time part month replac ahd head order night dr
op head damag job complet"
##
## $`2`
## [1] "decal generat call custom product id sec drive thru equip new replac pleas proceed propo
s order instal new headset nte turn ron freeman propos submit cpg pend approv propos declin"
##
## $`3`
## [1] "decal generat call custom product id sec teller undercount steel equip new pleas portabl
teller locker teller bus per propos nte michael per tech threatt assign tech kennedi wji per cas
ey hill schedul even per casey hill work complet close call complet"
##
## $`4`
## [1] "vat survey pleas indic call text quantiti vat lane deal drawer manufactur model pleas al
so indic vat video instal vat equip pleas specifi clear est call click ih updat inform regard pi
ctur ih close account rep et ds apxxcazcvatdieboldvisu auto tellerunlist compon detail notesunli
st compon modul debrief detailsequip function properlyequip oper error found problem found bill
call equip survey need go site site diebold commerci drawer vat way cctv lane contract"
##
## $`5`
## [1] "decal generat call custom product id jeff leechas com hi brian like schedul annual combo
chang marco island chase bank wednesday pleas confirm appoint decemb th brian harrison rakib saf
eti deposit lock adjust hing one lock chang combo test bank employe"
```

```
bill_notes_cleaned %>% lapply(function (doc) doc$content) %>% extract(1:5)
```

```
## $`1`
## [1] "NA"
##
## $`2`
## [1] "NA"
##
## $`3`
## [1] "pleas provid repair time bill thank send special queue account team review nonbil will b
ill"
##
## $`4`
## [1] "NA"
##
## $`5`
## [1] "piec equip combo chang ok bill key bill sr contract work"
```

```
billing_df$`Call Text` <- call_text_cleaned %>% sapply(function (doc) doc$content)
billing_df$`Billing Notes` <- bill_notes_cleaned %>% sapply(function (doc) doc$content)
billing_df$`Item Desc` <- item_desc_cleaned %>% sapply (function (doc) doc$content)
```

3. Tokenization

```
CONSTANTS <- list(
  # We will only consider the top 10,000 words in the dataset
  MAX_WORDS = 10000,
  # We will cut text after 100 words
  MAX_LEN = 100
)
```

3.1 Tokeninzing Categorical data

We will start off by encoding the labels of `Invoiced (Y/N)` using the `to_categorical` from keras

```
labels <- billing_df %>%
  use_series("Invoiced (Y/N)") %>%
  as.numeric() %>%
  subtract(1) %>%
  ifelse(("Invoiced (Y/N)" == "Y"),1) %>%
  as.array()

cat('Shape of label tensor:', dim(labels), "\n")
```

```
## Shape of label tensor: 2000
```

```
View(labels)
```

```
type <- billing_df %>%
  use_series("SR Type") %>%
  as.numeric() %>%
  to_categorical() %>%
  as.array()

cat('Shape of SR Type tensor:', dim(type), "\n")
```

```
## Shape of SR Type tensor: 2000 5
```

```
View(type)
```

3.2 Tokenizing Free Form Text

We will tokenize each free form text: `Call Text` , `Billing Notes` , and `Item Description` separately.

3.2.1 Call Text

We will start out by tokenizing `Call Text` :

```
call_text_df <- billing_df %>% select(c("Call Text"))
```

A `tokenizer` object will be created and configured to only take into account the 20,000 most common words, then builds the word index.

```
tokenizer <- text_tokenizer(num_words = CONSTANTS$MAX_WORDS) %>%
  fit_text_tokenizer(call_text_df$`Call Text`)
```

We then turn the texts into lists of integer indices

```
sequences <- texts_to_sequences(tokenizer, call_text_df$`Call Text`)
```

How you can recover the word index that was computed

```
word_index <- tokenizer$word_index

cat("Found", length(word_index), "unique tokens.\n")
```

```
## Found 4657 unique tokens.
```

```
call_text_data <- pad_sequences(sequences, maxlen = CONSTANTS$MAX_LEN)

cat("Shape of data tensor:", dim(call_text_data), "\n")
```

```
## Shape of data tensor: 2000 100
```

3.2.2 Billing Notes

We then tokenize `Billing Notes` :

```
billing_notes_df <- billing_df %>% select("Billing Notes")
```

```
tokenizer <- text_tokenizer(num_words = CONSTANTS$MAX_WORDS) %>%
  fit_text_tokenizer(billing_notes_df$`Billing Notes`)
```

```
sequences <- texts_to_sequences(tokenizer, billing_notes_df$`Billing Notes`)
```

```
word_index <- tokenizer$word_index

cat("Found", length(word_index), "unique tokens.\n")
```

```
## Found 302 unique tokens.
```

```
billing_notes_data <- pad_sequences(sequences, maxlen = CONSTANTS$MAX_LEN)

cat("Shape of data tensor:", dim(billing_notes_data), "\n")
```

```
## Shape of data tensor: 2000 100
```

3.2.3 Item Description

Finally we tokenize `Item Desc` :

```
item_desc_df <- billing_df %>% select(c("Item Desc"))
```

```
tokenizer <- text_tokenizer(num_words = CONSTANTS$MAX_WORDS) %>%
  fit_text_tokenizer(item_desc_df$`Item Desc`)
```

```
sequences <- texts_to_sequences(tokenizer, item_desc_df$`Item Desc`)
```

```
word_index <- tokenizer$word_index

cat("Found", length(word_index), "unique tokens.\n")
```

```
## Found 261 unique tokens.
```

```
item_desc_data <- pad_sequences(sequences, maxlen = CONSTANTS$MAX_LEN)

cat("Shape of data tensor:", dim(item_desc_data), "\n")
```

```
## Shape of data tensor: 2000 100
```

```
save.image(file = "data/data_prep_workspace.RData")
save(billing_df, CONSTANTS, call_text_data, billing_notes_data, item_desc_data, type, file="dat
a/data_preparation.RData")
```