1 Installing hedonicText

Using **devtools**, install **hedonicText**

```
library(devtools)
install_github("browak/NowakSmith/hedonicText" , quiet=T)
library(hedonicText)
```

hedonicText cannot be installed using CRAN.

2 Data

hedonicText comes with a data set of 24,446 property transactions with sale price, square footage, and remarks. The remarks are text descriptions of the proeprty created by the listing agent. The data set can be loaded using

```
data(listings)
listings <- listings[1:1000, ]
str(listings)

## 'data.frame': 1000 obs. of  3 variables:
## $ price : num 106300 65000 255000 129900 180000 ...
## $ sqft : int 1424 724 3040 1583 2571 2059 1379 1870 1870 2769 ...
## $ remarks: chr "RARE 4 BEDROOM HOME WITH FRESH PAINT IN & OUT. NEW LIGHT CARPET
##    AND VINYL PLUS DUAL PANE WINDOWS. TAXES ARE INV"| __truncated__
##    "********************HOUSE IS VACANT - ON
##    LOCKBOX!!!!!!!********************** ************************OLDER "|
##    __truncated__ "YOU'LL NEVER NEED MORE SPACE THAN THIS NEWER LIGHT & NEUTRAL
##    EXECUTIVE HOME ON OVER 1/4 ACRE WITH ULTRA SPACIOU"| __truncated__ "GREAT
##    REMODEL WITH TONS OF FEATURES! KITCHEN REMODELED 3 YRS AGO ALONG WITH NEW
##    HEAT/AC. ROOF 1 YR, WATERHEATER"| __truncated__ ...
```

3 Functions

The remarks in the data section are all capitalized, include letters, numbers, and non alpha-numeric characters. **hedonicText** includes a basic function, **cleanText**, to clean the text. Of course, you can always clean the text yourself.

```
cleanRemarks <- cleanText(listings$remarks)

## Loading required package:  tm
## Loading required package:  NLP

str(cleanRemarks)

## chr [1:1000] "rare bedroom home fresh paint new light carpet vinyl plus dual
##    pane windows taxes investors seller never occupied home" "house vacant
##    lockbox older home huge lot sold condition" "ll never need space newer light
##    neutral executive home acre ultra spacious rooms thruout rare car garage
##    real p"| __truncated__ "great remodel tons features kitchen remodeled yrs
##    ago along new heat ac roof yr waterheater softner huge loft b"|
##    __truncated__ ...
```

By default, **cleanText** will convert all text to lowercase, remove numbers, remove punctuation, remove stop words, and remove single letters. The list of stop words comes from the list of stopwords in the **tm** package.

You can also identify the most frequent n-grams in the data. These n-grams can be used to tokenize the text. This is done using the **flexGramCount** function. The most frequent unigrams are found using

```
unigramCount <- flexGramCount(cleanRemarks , maxN=1 , minN=1)

## Loading required package:  ngram
## Loading required package:  doParallel
## Loading required package:  foreach
## Loading required package:  iterators
## Loading required package:  parallel
## Loading required package:  stringi

head(unigramCount)

##    ngrams freq N
## 1    home 1028 1
## 2    room  719 1
## 3 kitchen  687 1
## 4    tile  489 1
## 5     new  489 1
## 6   great  486 1
```

Bigrams contain more detailed information but occur less frequently

```
bigramCount <- flexGramCount(cleanRemarks , maxN=2 , minN=2)
head(bigramCount)

##             ngrams freq N
## 1      family room  214 2
## 2    covered patio  190 2
## 3     ceiling fans  176 2
## 4     ceramic tile  119 2
## 5 vaulted ceilings  107 2
## 6       floor plan  101 2
```

Trigrams contain even more information but occur even less frequently

```
bigramCount <- flexGramCount(cleanRemarks , maxN=3 , minN=3)
head(bigramCount)

##                  ngrams freq N
## 1            cul de sac   67 3
## 2            de sac lot   35 3
## 3       open floor plan   28 3
## 4    tile entry kitchen   28 3
## 5     dual pane windows   26 3
## 6 fireplace family room   25 3
```

You can also identify a blend of n-grams for various n using

```
flexgramCount <- flexGramCount(cleanRemarks , maxN=5 , minN=2)

## [1] "no 5-grams found with frequency larger than minCount. Skipping to 4-grams"

head(subset(flexgramCount , N==2) , n=3)

##           ngrams freq N
## 53  ceiling fans  176 2
## 54 covered patio  139 2
## 55   family room  111 2

head(subset(flexgramCount , N==3) , n=3)

##              ngrams freq N
## 6        cul de sac   32 3
## 7   open floor plan   28 3
## 8 dual pane windows   26 3

head(subset(flexgramCount , N==4) , n=3)

##                           ngrams freq N
## 1                cul de sac lot   35 4
## 2 vaulted ceilings plant shelves   13 4
## 3      leave message use lockbox   10 4

head(subset(flexgramCount , N==5) , n=3)

## [1] ngrams freq   N
## <0 rows> (or 0-length row.names)
```

4 Hedonic Models

Using a given set of tokens, you can estimate a hedonic pricing model. First, you create a matrix of indicator variables for each of the tokens

```
tokenList <- bigramCount$ngrams
M <- tokenMatrix(cleanRemarks , tokenList)

## Loading required package:  Matrix
## Error in {:  task 1 failed - "object 'text2' not found"

head(colnames(M))

## Error in is.data.frame(x):  object 'M' not found

str(M)

## Error in str(M): object 'M' not found
```