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 property created by the listing agent. The data set can be loaded using

```
data(listings)
listings <- listings
str(listings)

## 'data.frame': 24446 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 [1:24446] "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 26218 1
## 2    room 16375 1
## 3 kitchen 15841 1
## 4     new 13304 1
## 5    tile 12422 1
## 6   great 11942 1
```

Bigrams contain more detailed information but occur less frequently

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

##             ngrams freq N
## 1    covered patio 5229 2
## 2     ceiling fans 4591 2
## 3      family room 4405 2
## 4     ceramic tile 3029 2
## 5 vaulted ceilings 3016 2
## 6       car garage 2407 2
```

Trigrams contain even more information but occur even less frequently

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

##                ngrams freq N
## 1          cul de sac 1421 3
## 2   dual pane windows  702 3
## 3 large covered patio  684 3
## 4     open floor plan  680 3
## 5          de sac lot  668 3
## 6   great curb appeal  516 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
## 35     ceiling fans 4261 2
## 36    covered patio 4086 2
## 37 vaulted ceilings 2695 2

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

##                ngrams freq N
## 3          cul de sac  749 3
## 4   dual pane windows  702 3
## 5 large covered patio  684 3

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

##                            ngrams freq N
## 1                cul de sac lot  664 4
## 2 vaulted ceilings plant shelves  321 4
```

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 <- unigramCount$ngrams
M <- tokenMatrix(cleanRemarks , tokenList)

## Loading required package:  Matrix

str(M)

## Formal class 'ngCMatrix' [package "Matrix"] with 5 slots
## ..@ i : int [1:1620657] 0 1 2 3 4 7 8 9 12 13 ...
## ..@ p : int [1:790] 0 17019 32297 45084 54971 66866 76258 84822 94530 103553
##    ...
## ..@ Dim : int [1:2] 24446 789
## ..@ Dimnames:List of 2
## .. ..$ : NULL
## .. ..$ : chr [1:789] "home" "room" "kitchen" "new" ...
## ..@ factors : list()
```

The tokens matrix can be used with additional regressors in a hedonic model. The additional regressors are not penalized in the penalized regression. The model is then estimated using a LASSO with a cross-validated penalty term.

```
y <- log(listings$price)
X <- cbind(listings$sqft , listings$sqft**2)
unigramFit <- hedonicWithText(y , X , M)
```

```
## Loading required package:  glmnet
## Loaded glmnet 2.0-13
```

The coefficients for the tokens in the penalized regression can be extracted using

```
bFit <- coef(unigramFit , s='lambda.min')
```

tokenTable will transform coefficients into a data frame for presentation

```
tokenDataFrame <- tokenTable(bFit)
head(tokenDataFrame , n=10)

##         token coefficient
## 1     granite       0.155
## 2         bid      -0.125
## 3  scottsdale       0.123
## 4  travertine       0.086
## 5        acre       0.085
## 6       views       0.083
## 7        game      -0.076
## 8        fees      -0.074
## 9         fix      -0.068
## 10      horse       0.067
```